

BEEBUG

Issue 1
Number 1
April 1986

Beebug News

for BBC
micro users



MIXED GRAPHICS AND TEXT

"GRAFTXT" is a program which enables you to mix both graphics and text together on the same page. It uses the printer's own text characters to print the text and uses a flexible mode 4 screen dump to print the graphics. In this way, the number of text fonts depends on your printer. It can be italics, pica, elite or near letter quality, like the print seen here. The choice is yours. It allows you to create your own page format and save it. A suitable mode 4 picture may then be cropped and printed together with your text defined in your format. The result?...take a look at this page!

You may print large pictures, up to nearly full size screen in mode 4, or small pictures, down to one character in size. You may place the picture anywhere on the page, at either left or right ends, or in the middle between text. You may choose to print your pictures



with borders around them or without. You may justify text left, right or centre, and even combine sections of text and graphics together to produce a full length page like this one. And all done with the greatest of ease!

GRAPHICS and TEXT

"GRAFTXT" is written part contains the design the screen page. This has a WYSIWYG format designer you get'. You select the graphics area, type in before saving it on disc. controls the output to previously created format load in a mode 4 picture,

Of course, every piece limitations and this is facilities provided here useful printer's tool and

GRAPHICS and TEXT

TEXT TEXT
TEXT TEXT

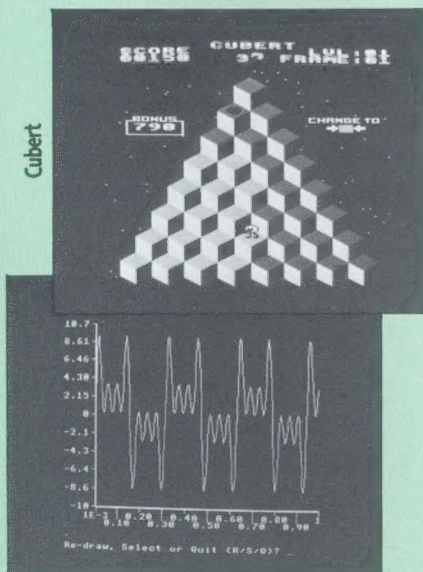
in two parts. The first program, using mode 0 as been scaled to give you (ie. 'what you see is what size and position of your the text and justify it. The second part, "GRAFT2", the printer. You load in a and select the font. Then crop it, and print it out.

of programming has its own no exception. However, the will no doubt make it a will have many uses.



You can use it for printing letter heads, business and greeting cards, newsletters, or simply just to decorate your social mail and impress your friends. Of course, it may not be as good as the AMX Pagenaker or Fleet Street Editor by Mirrorsoft, but then you save on quite a bit of cash, and the results aren't all that bad. Perhaps you may even like to write your own mixed text and graphics program - well, why not join BEEBUG now and read all about printer graphics in a forthcoming article. HAPPY PRINTING!

BEEBUG PAGE DESIGNER



Music 5000

Merge, Part-Merge and Part Save	9
Business Graphics (Part 1)	12
Multi-Column Wordwise Plus	19
Printer Graphics (Part 3)	
BEEBUG Page Designer	21
First Course — Character Formation	38
The Master Series	
Executive Power	42
Master Hints	45
Master Font Manager	46
BEEBUG Workshop — Date Handlers Part 2	52
Digital Storage Oscilloscope (Part 2)	48
Cubert	58

Hybrid's Music 5000	6
Micro Trader	16
Games Reviews	26
Pursuing the Trivial	
Pyscastia	
Galaforce	
Sideways RAM Exposed	41
Adventure Games	51
Watford ROM/RAM Board	54

Editorial Jottings	3
News	4
BEEBUGSOFT Forum	5
Supplement	28-37
Postbag	56
Hints and Tips	57
Points Arising	61

GENERAL	Accurate Wordwise
Graphics Turn On	Readable Programs
ADFS File Names	MASTER
*EXEC Nesting	Easy DFS CAT from ADFS
Screen & Window Width	Co-resident Editor and Program
Wordwise Plus Zeros	Executive Editor
Lives, Lives & More Lives	Editor Program Loss
File Data Storage	ADFS *ACCESS

BEEBOW COMPUTERS					
Account No.		Statement Date		Date	
1000		15.09.86		15.09.86	
Value	Date	Invoice No.	Invoice Value	Invoice Value	Net
Receipts					
10.00.00	10.00.00	100.00	100.00	100.00	100.00
10.00.00	10.00.00	100.00	100.00	100.00	100.00
10.00.00	10.00.00	100.00	100.00	100.00	100.00
Statement Totals					
100.00		100.00		100.00	
Settlement Discount					
10.00.00		10.00.00		10.00.00	
Net Amount Due					
10.00.00		10.00.00		10.00.00	
BEEBOW COMPUTERS					
Account No.		Statement Date		Date	
1000		15.09.86		15.09.86	
Settlement Discount					
10.00.00		10.00.00		10.00.00	
Net Amount Due					
10.00.00		10.00.00		10.00.00	

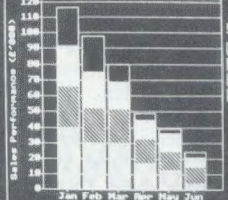
```

200UNTIL OptS="Q"
210CLS
220END
230
1000DEF PROCInit
1010maxpts=2000
1020IN DataC(maxpts),osclbuf 30
1030IN:0 P=1:FX16.1
1040ENDPROC
1050
1060DEF PROCMenu
1070CLS:COL:0:129:COLOUR 0
1080PRINTTAB(0,2) "DIGITAL OSCILLOSCOP
E
1090PRINTTAB(16,4) "OPTIONS"
1100PRINTTAB(10,8) "M"TAB(10,15) "O"TAB(1
0,17) "T"TAB(10,20) "S"
1110IF M THEN PRINTTAB(10,10) "P"TAB(10
,13) "S"
Escape at line 1110
>SPRVE FRED,1000,1040
>MERCE FREDI,1200,1280

```

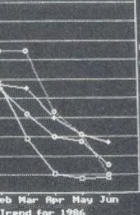
Demonstration Data

BBC Micro Business Graphics



Demonstration Data

BBC Micro Business Graphics



EDITORIAL JOTTINGS

SUPPORT FOR THE MASTER

In this issue of BEEBUG we have expanded our coverage of the Master series from four to six pages to allow us to publish even more programs, articles and other information on this system. Compact users will also find that our own 'Master Series' is full of interest to them too.

We have achieved this by reducing the size of the supplement, rather than by any reduction in editorial content. And this in no way diminishes our support for owners of the model B and B+. In fact, one of our Master articles, on the use of EXEC files, contains ideas that are applicable to all Beeb owners.

TRADE IN YOUR BEEB

Many BEEBUG members have expressed interest in upgrading their model B to a Master 128 or Master Compact. Last month we were able to announce a trade-in scheme which enables members to obtain a substantial trade-in allowance on their old machine. In addition, all Model B micros obtained by us are fully tested, and offered for resale in guaranteed full working order. So, if you want to upgrade your system, or purchase a model B at a very reasonable price, see the BEEBUG mail order price list enclosed with this issue.

REVISED PROGRAM CLASSIFICATION

The growing variety of Acorn systems has caused us to revise completely the range of symbols that we use to classify programs, and where appropriate, reviews, published in the magazine. The main changes are the inclusion of separate disc symbols for the DFS and the ADFS, and a symbol for the new Master Compact which distinguishes between ROM Basic VI and disc Basic IV. The complete set of symbols is shown below. The new symbols show clearly the valid combinations of machine (version of Basic) and filing system for each item (and Tube compatibility). Remember that a single line through any symbol indicates partial working (normally a few changes will be needed) and a cross over a symbol shows total unsuitability. For reviews we do not distinguish between Basic I and Basic II, but merely put a 'B', for the model B.

COMPUTER SYSTEM

Master (Basic IV)

Compact (Basic VI)

Compact (Basic IV)

Model B (Basic II)

Model B (Basic I)

Electron

FILING SYSTEM

ADFS

DFS

Cassette

TUBE COMPATIBILITY

Tube



News News News News News News News

Acorn Profitable

Acorn has reported a profit of nearly £300,000 in the six months to June, against a loss of nearly £100,000 in the previous half year. It looks as though Acorn is now truly out of the doldrums and that the future of the BBC micro in all its forms is now as secure as it can be.

Free Ultracalc Utilities

BBC Soft is giving away utilities for the Ultracalc spreadsheet ROM absolutely free. A formatted disc sent to BBC Software, 33 Marylebone High Street, W1M 4AA with stamps for return postage will be returned with graph drawing, Acorn Database to Ultracalc data conversion, printing and other utilities.

Ted 4 WE

Watford Electronics has released a mode 7 teletext editor called Ted. Supplied on a 32K ROM, Ted allows the creation of screens in the viewdata format, or as part of a large screen with one screen-sized window on the screen at a time. Ted costs £35 from Watford on 0923-37774.

New Epson Printer

The popular Epson LX-80 NLQ printer has been replaced by the LX-86. The new machine is very similar to the old, but can operate faster. Epson claims the LX-86 is 20% faster than the old model, capable of 120 characters per second in draft mode. The recommended price for the LX-86 is £275. Epson is on 01-902 8892.



Spider Control

The Spider hardware and software real time control package (see BEEBUG news Vol.4 No.9) has been updated to Spider2, and a cartridge version is available for the Master. Spider2 can deal with all kinds of digital and analogue inputs and outputs with extensions to Basic to control the board. Prices start at £65 from Paul Fray Ltd. on 0223-66529.

Add-On NLQ

M & M Technix has good news for owners of printers without NLQ. The NLQ printer buffer plugs into any Epson compatible printer and connects to a Beeb. Anything sent to the printer is converted by the unit to NLQ in one of three fonts in the device. The fonts may be chosen from 17 produced by M&MT. The unit is also a printer buffer and is available with 8K or 32K buffer sizes. Prices start at £125. Further details from M&MT on 0332-381649.

3D Updated

Design Dynamics' Interactive 3D package (see BEEBUG Vol.4 No.10) has been updated to include automatic circle drawing. The new version can save screens to disc for dumping to a printer not covered by the software's built-in FX80/Taxan screen dump. The price remains £8.95 (tape), £12.95 (disc). Details from Design Dynamics on 0525-402447.

Beeb Link-up

Dinosaur Software has released two music-making packages that will conduct two BBC micros in harmony. Musicpen is a light-pen driven music composition program on ROM which enables Music-Link to use all eight sound channels of two BBC micros connected together via the RS423 connectors. Musicpen costs £28.75 and Music-Link £9. Further information from Dinosaur on 0223-322244.

Acornsoft/Superior Game

The latest game to come out of the alliance between Superior Software and Acornsoft is a battlefield game called Stryker's Run. The game involves fighting your way across a battlefield, shooting enemy soldiers and commandeering any abandoned planes or tanks. It costs £9.95 on tape and £11.95 on disc (£14.95 on 3.5 inch disc). Superior Software is running a competition for purchasers of the game with over £200 of prizes. Superior Software is on 0532-459453.



Hybrid's Music 5000

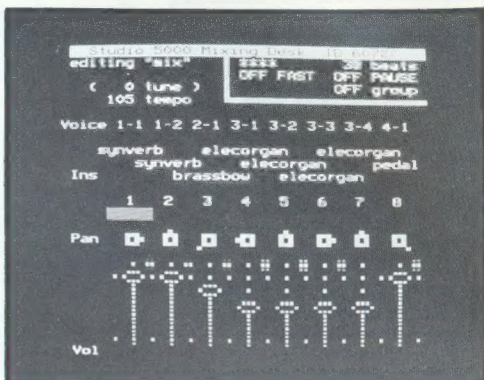
At last Hybrid Technology has released its successor to the acclaimed Music 500. It is called the Music 5000, but is it ten times as good? Ian Waugh, author of books and articles on the subject of music and micros, conducts his own review.

Product : Music 5000
Supplier : Peartree Computers Limited,
 St. George's House,
 14 George Street,
 Huntingdon,
 Cambridgeshire PE18 6BD.
 (0480) 50595
Price : £159.95

Product : Music 500 Upgrade Pack
Supplier : Hybrid Technology Limited,
 Unit 3, Robert Davies Court,
 Nuffield Road,
 Cambridge CB4 1TP.
 (0223) 316910
Price : £69.00

The original Acorn Music 500 Synthesiser was launched over two years ago and reviewed in May 1985 in BEEBUG Vol.4 No.1. It could produce up to eight voices using 16 channels of sound and its enormous range of sounds rivalled and, in many cases, surpassed those of dedicated synthesisers costing two and three times its asking price of £199. It was also unique in being totally software driven by the Beeb, and it promised to open up a whole new area of music to Beeb owners.

If the Music 500 had any drawbacks it was its own sheer wealth of possibilities. It was controlled (i.e. programmed) in a language called AMPLE (Advanced Music Production Language Environment) and in order to get the best out of the system you really had to learn the language. The manual, however, was not really up to the task, and there must be many Music 500 owners who struggled to understand what is, after all, a new, very powerful and sophisticated language.



Finally, after lots of rumours, the Music 5000 appeared at the Acorn User Show. To quell the speculation, the facts are these: there is only one hardware difference between the 5000 and the 500 and that lies in the 1 MHz bus. This has been re-designed to make it compatible with nearly all BBC computers: the BBC B (with OS 1.2), B+, B+128 and the Master 128 (but not Compact). The Music 500 was designed specifically for the BBC B and is not guaranteed to work with any other model (but my 500 works quite well with my Master). The Music 500 has now been discontinued, although you may find one second hand, but try it with your computer before you buy.

Thus the main difference between the 5000 and the 500 lies in the software, and the Music 500 Upgrade Pack consists of an AMPLE ROM and a new User Guide. This will give you all the software features of the new 5000 but it will not make a previously incompatible 500 suddenly compatible.

The basic unit of an AMPLE program is the word. A word can be defined to do almost anything. For example, it can hold instructions to play a sequence of notes, set tempo, transpose or call up a new instrument. Although the old software gave you a few predefined waveforms and envelopes you really had to start from scratch to produce anything significant.

The difference in software is immediately apparent upon booting the system disc. A whole new front-end called the Studio 5000 has been designed containing many features to simplify the entry of music and the design of

instruments. The main menu offers: Run Program (run the program resident in memory), Notepad (a music editor and instrument designer), Staff Editor (for editing music on the staff), Mixing Desk (for mixing and balancing the different parts of a piece), and several utilities.

The design philosophy behind the new presentation is one of software extension, whereby modules can be loaded into the system as and when required. Some modules, such as instrument definitions, are loaded permanently when the system starts; others, such as the Notepad and Staff Editor, are only loaded when required. Thus new modules can be loaded and used as simply as you would plug in a new ROM.

The manual gets off to a good start by telling you how to call up and play the excellent demo tunes. It then moves on to describe the Mixing Desk which is one of the new front-end modules. This gives you a mode 7 display of an eight-channel mixing desk, one channel per voice. It goes further than a conventional mixing desk - it is computerised, after all. It enables you to alter the stereo position of each voice, adjust the volume and alter the instrument the voice is using. The instruments are chosen from the 14 preset ones or any others which may have been defined especially for the piece. You can make adjustments as the music is playing so that you can see and hear how your mix sounds in the context of a tune.

There are other effects on the Mixing Desk, too. You can transpose the piece, alter the tempo, pause the music and fast wind through it. You can save different mixes and recall them at will. It's a computerised mixing console, easy to use, fun and practical at the same time.

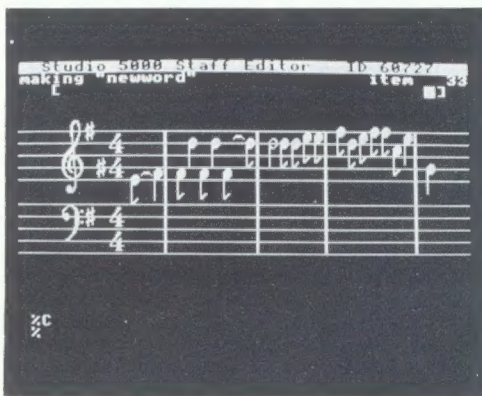
The second module the manual introduces is the Staff Editor. For traditional musicians, this lets you enter music on a staff or stave. Note entry and editing is accomplished with the cursor keys and judicious use of the Shift key, a process not completely unlike System's The Music System. Unlike TMS, however, the AMPLE Editor has provision for almost every conceivable musical nuance you can imagine - and then some you can't.

You can enter the full range of notes, sharps, flats, accidentals, rests, ties,

slurs, triplets and duplets. You can also set non-standard key signatures, define note style (e.g. legato or staccato), set dynamics (loudness) and accent notes. You can insert bar lines (or not) and if you use a time signature (not compulsory), the program will check the length of the bars. You can also define and insert your own instruction words relating to performance.

You can enter chords on the staff, too. The notes are staggered across the stave rather than appearing in a vertical line. It looks odd at first but it's easy to adjust to. You can also assign percussion sounds to six notes in the treble clef and create your own rhythms.

The Staff Editor is ideal for musicians who feel most comfortable working with traditional notation. It will also help anyone entering music from a written score and, of course, you can see at a glance what the piece looks like. This is a very powerful addition to the AMPLE language.



The Notepad is a multi-purpose editor that handles data in text form, unlike the graphic displays of the other two editors. The main part of the screen is defined as a text window and you can move to any position in it with the cursor keys. New characters overwrite what is on the screen, and function keys are used to insert and delete lines and characters. The creation and editing of instructions is therefore very easy. The Notepad has two main uses: to create music and to define instruments. A function key switches between the two applications.

AMPLE has a music notation of its own (already familiar to Music 500 users), and the manual is keen to stress how much simpler and more logical it is than the staff notation. A note length and octave number are set first of all and then notes are entered as a series of note names. You can add accidentals, define key signatures, play chords and do all the musical things you want to do. This notation is, in fact, the system around which the other editors were designed, and you can edit pieces here that were created with the other editors. The Notepad is undoubtedly the most powerful of the three editors and some facilities can only be used from within it.

Echo is a built-in special effect and you can adjust the echo parameters to produce anything from reverberation to a standard tape echo effect. The repeating echoes can be detuned and transposed and given individual stereo positions in the Mixing Desk. Thus you could, for example, produce eight echoes which dropped in both pitch and volume as they passed from one speaker to the other. You can also program gradual changes in tempo (rallentando and accelerando) and volume (from ppp to fff and vice versa), a facility virtually unknown on computer based music systems. The dynamic level control can be redirected to produce autopanning and slide effects. Such is the versatility of software based controls.

The Notepad is also used to create instruments. Unlike the 500 software, you no longer have to create waveforms and pitch and amplitude envelopes - these have already been done for you. You can choose from 14 waveforms, 17 amplitude envelopes and 17 pitch envelopes. It's not as though you only have 14 x 17 x 17 possible combinations, either, because you can add synchronisation, and ring and frequency modulation.

Most instruments are designed using one or two of the 16 available sound channels, but you can use up to all 16 on one voice if you wish. Individual voices can be detuned, transposed and given separate stereo positions and volume levels. You can even call up an instrument sound and apply further modifications to

it. The User Guide includes diagrams of all the waveforms and envelopes and gives details of the predefined instruments along with a further 25 for you to type in. The creation and editing of instruments has been simplified enormously by the Notepad Editor.

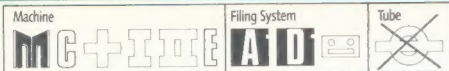
Unlike the Music 500, you no longer have direct control over waveform and envelope design and some of the other little niceties which made the AMPLÉ language so difficult for newcomers to grasp. I doubt, actually, that they will be missed but a waveform and envelope editor is under development, and it will be easy to load this into the program using the module system. *

The new User Guide is a great improvement over the old one and you will be able to work through the front-end editors immediately. With just a little effort you should be able to produce pieces in hours, not days. A little musical knowledge, however, will not go amiss. The manual, therefore, generally does an excellent job although even more information would have been welcomed. As you delve deeper into the language some questions may arise which it won't answer. To take advantage of all the language's programming facilities, an AMPLÉ Programmer's Guide is being written. For full integration with established music systems, however, we still await the MIDI interface and keyboard.

If you're a Music 500 owner and have struggled with AMPLÉ programming then the Upgrade Pack is a must. It will put new life into your system and, as you will already have some idea of how AMPLÉ works, the whole process should be even easier for you.

As a powerful musical tool and synthesiser, the Music 5000 can only be complimented and is certainly recommended. No other instrument in this price range comes close to offering both the sounds and the programming flexibility of this device and the new front-end makes it easy to use, too. If you need more convincing try the demo audio cassette from Peartree at £2.95.

B



Merge, Part-merge and Part-save

Tired of deleting, SPOOLing and EXECing? This utility from Jagdish Sah gives your micro two powerful * commands, *MERGE and *PSAVE.

The User Guide describes two methods of merging a Basic program from a tape or disc file onto a program in memory. The first method is based on *SPOOLing the program to be merged in a file which is then *EXECed into the current program. Although it achieves the objective adequately, the method is somewhat long-winded.

The second method involves *LOADing the file to be merged at the end of the current program. Thus, it succeeds only in appending one program to another.

Obviously there is a need for a utility which will extract the required lines from a file and merge them into the program in memory without going through the rigmarole described above. The accompanying listing provides a new command, *MERGE, which will serve this purpose. In addition it also implements a second command, *PSAVE, which will save any part of the program in memory.

USING THE UTILITY

Type in the program as listed. Save it to tape or disc before you run it. After running, save the machine code routine produced with the help of the displayed command. If a 'Checksum error' message is displayed, check the program carefully for any mistakes. When you intend to use the utilities, type:

```
*RUN MERGEMC
(or *MERGEMC for disc users). This will
set up both commands ready for use. The
*MERGE command has the syntax:
*MERGE <filename>,<startline>,<endline>
```

It will extract all lines between the startline and the endline (inclusive) from the specified file and merge them into the program in memory. New lines from the file will replace the old ones in the memory if

they have the same line numbers. A typical example of the command is:

```
*MERGE OLDPROG,1000,1999
```

This will merge all lines between 1000 and 1999 (inclusive) from the file OLDPROG.

```
*MERGE OLDPROG,0,1999
```

will merge all lines from the beginning of the file to the line 1999.

```
*MERGE OLDPROG,0,0
```

will merge the whole file. The routine replaces the 0 for the endline with 32767, the largest possible line number. In the examples above OLDPROG may be replaced by any valid file name. For tape files it may also be replaced by "" which, as usual, means the next file on the tape.

The *PSAVE command has the syntax:

```
*PSAVE <filename>,<startline>,<endline>
```

This will save all lines between the startline and the endline (inclusive) from the program in memory into a new file. The line numbers carry the same meaning as for the MERGE command. But in this case they pertain to the program in memory.

If the Escape key is pressed during the merge or partsave operation, everything is tidied up before the routine is exited. In the merge operation this means inserting the current line properly, so that the program in memory is not corrupted, before closing the input file. In the partsave operation, the current line together with the end-of-program marker is sent to the output file before the latter is closed.

ERROR MESSAGES

The program also detects and reports on any errors that may arise. The routine first checks whether a valid program is present at PAGE in memory. If not, a "Bad program" message is displayed. Thus, if you want to use the *MERGE command to effect a part-load prior to a typing-in session remember to issue a NEW command first.

If an improper line number is specified, a "Bad Arg" message is output. A "Can't open file" message will be produced only on a disc system. For the merge operation it means that the specified file was not found on the disc.

For the partsave operation it may result from the disc or the catalogue being full.

PROGRAM NOTES

&900 is the start address for the machine code routine. If you find this inconvenient, a different value may be chosen by altering line 100. But do so only after the program has run once without giving a "checksum error" message. The machine code requires 3 pages in total, so for disc users, the code would occupy pages &9, &A and &B.

Tape users should note that the area &900-&AFF cannot be used - the routine needs it for BGET and BPUT operations- and should set code to &B00 in line 100.

The two new commands will work in immediate mode only. They will work with the cassette as well as the disc filing systems. The program is assembled relative to the Basic in your machine, and cannot be transferred to another machine with a different Basic without being re-assembled.

```

10 REM PROGRAM MERGE and PARTSAVE
20 REM VERSION B0.18
30 REM AUTHOR Jagdish Sah
40 REM BEEBUG November 1986
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :
100 code=&900
110 PROCassemble:PROCchecksum
120 IF S%>chksum%:PRINT"Checksum err
or":END
130 PRINT""Type""*SAVE MERGEMC ";~c
ode;" ";~P%
140 PRINT"to save the machine code pr
oduced,%"
150 END
160 :
1000 DEF PROCassemble
1010 REM Syntax:
1020 REM *MERGE <fsp>,<start>,<end>
1030 REM *PSAVE <fsp>,<start>,<end>
1040 osfind=&FFCE:osbget=&FFD7
1050 osbput=&FFD4:osbyte=&FFF4
1060 osargs=&FFDA:osfile=&FFDD
1070 page=&18:cmdptr=&19:ycmd=&1B
1080 newno=&2A:lineptr=&70:start=&72
1090 end=&74:handle=&76:savey=&77
1100 newlen=&78:temp=&79:stack=&7A
1110 pblk=&7E:anycom=&F8
1120 IF ?&8015=ASC("1"):RESTORE 2750:EL
SE IF ?&8015=ASC("2"):RESTORE 2780:ELSE
IF ?&8015=ASC("4"):RESTORE 2810:ELSE RES
TORE 2840

```

```

1130 READ insline,chkprog,getnumB
1140 READ getcharB,basic,chksum%
1150 :
1160 FOR opt=0 TO 2 STEP 2
1170 P%=code:[OPT opt
1180 LDA &209:CMP #newrtn DIV 256
1190 BEQ out:STA oldrtn2+2
1200 LDA &208:STA oldrtn2+1
1210 LDA #newrtn MOD 256
1220 SEI:STA &208:LDA #newrtn DIV 256
1230 STA &209
1240 CLI
1250 .out RTS
1260 .oldrtn LDX cmdptr
1270 LDY cmdptr+1:PLA:PLP
1280 .oldrtn2 JMP 0
1290 :
1300 .getval
1310 JSR getnumB:BEQ badarg:BMI badarg
1320 BIT&F8:BMINocom:JSR getcharB:CMP #
ASC(",")
1330 BNE badarg:.nocom:LDA newno+2
1340 ORA newno+3:BNE badarg
1350 LDX newno+1:BMI badarg
1360 LDA newno:RTS
1370 .badarg BRK
1380 OPT FNequb(&FE)
1390 OPT FNequs("Bad Arg")
1400 OPT FNequb(0)
1410 :
1420 .newrtn
1430 PHP:PHA:STX cmdptr:STY cmdptr+1
1440 JSR chkprog:LDY #0:STY lineptr
1450 .chkcmd LDA cmd,Y:BEQ merge
1460 CMP (cmdptr),Y:BNE chk2
1470 INY:BNE chkcmd
1480 .chk2 LDY #0
1490 .chkcmd2 LDA cmd2,Y:BEQ partsave
1500 CMP (cmdptr),Y:BNE oldrtn
1510 INY:BNE chkcmd2
1520 :
1530 .merge
1540 LDA #&40:JSR init
1550 .merge2 JSR getbyte:CMP #&0D
1560 BNE close:JSR getbyte
1570 STA newno+1:JSR getbyte
1580 STA newno:JSR getbyte
1590 STA newlen:LDA newno
1600 CMP start:LDA newno+1
1610 SBC start+1:BCS merge3:LDX #4
1620 .skipline JSR getbyte:INX
1630 CPX newlen:BNE skipline
1640 BEQ merge2
1650 .merge3 LDA end:CMP newno
1660 LDA end+1:SBC newno+1
1670 BCC close:LDX #4
1680 .getall JSR getbyte:STA &700,X
1690 INX:CPX newlen:BNE getall
1700 LDA #&0D:STA &700,X
1710 JSR setbasic:LDY #4

```



```

1720 JSR insline:CLC:BCC merge2
1730 :
1740 .close
1750 LDA #0:TAY:JSR osfind
1760 .close2 LDX stack:TXS
1770 JSR setbasic:JMP basic
1780 :
1790 .getbyte
1800 LDY &FF:BMI close
1810 LDY handle:JSR osbget
1820 BCS close:RTS
1830 :
1840 .partsave
1850 LDA #80:JSR init
1860 .psave2 LDY #2
1870 LDA (lineptr),Y:CMP start
1880 DEY:LDA (lineptr),Y
1890 SBC start+1:BCS psave3
1900 JSR inclineptr:BCC psave2
1910 .psave3 LDA &FF
1920 BMI exitpsave:LDY #2
1930 LDA end:CMP (lineptr),Y:DEY
1940 LDA end+1:SBC (lineptr),Y
1950 BCC exitpsave:LDY #3
1960 LDA (lineptr),Y:STA temp
1970 LDY #0:STY savey
1980 .psave4 JSR putbyte
1990 LDY savey:CPY temp:BNE psave4
2000 JSR inclineptr:BCC psave3
2010 .exitpsave LDA #&0D
2020 JSR putbyte2:LDA #&FF
2030 JSR putbyte2:LDA #0:TAY
2040 JSR osfind:LDA #0:TAY
2050 JSR osargs:CMP #4:BNE exitps2
2060 LDX #pblk MOD 256
2070 LDY #pblk DIV 256
2080 LDA #1:JSR osfile
2090 .exitps2 JMP close2
2100 :
2110 .inclineptr
2120 LDY #3:LDA (lineptr),Y
2130 CLC:ADC lineptr:STA lineptr
2140 BCC inclineptr2:INC lineptr+1
2150 .inclineptr2 CLC: RTS
2160 :
2170 .putbyte
2180 LDY savey:INC savey
2190 LDA (lineptr),Y
2200 .putbyte2 LDY handle:JMP osbput
2210 :
2220 .setbasic
2230 LDA #BB:LDX #0:LDY #&FF
2240 JSR osbyte:TXA:TAY:LDA #&97
2250 LDX #&30:JMP osbyte
2260 :
2270 .init
2280 PHA:STY ycmd:LDY#0:STYanycom
2290 LDA ycmd:CLC:ADC cmdptr
2300 TAX:STX pblk:LDA cmdptr+1
2310 ADC #0:TAY:STY pblk+1

2320 LDYycmd:.retlp:LDA(cmdptr),Y:INY
2330 CMP#ASC(",") :BNEretlp
2340 LDA#13:STYycmd:DEY:STA(cmdptr),Y:L
DY pblk+1
2350 PLA:JSR osfind:TAX:BNE opened
2360 BRK:OPT FNequb(200)
2370 OPT FNequs("Can't open file")
2380 OPT FNequb(0)
2390 .opened STA handle:LDYycmd:JSR get
val
2400 STA start:STX start+1:DECanycom
2410 JSR getval:STA end:STX end+1
2420 ORA end+1:BNE notzero:LDA #&FF
2430 STA end:LDX #&7F:STX end+1
2440 .notzero LDA page
2450 STA lineptr+1:TSX:DEX:DEX
2460 DEX:DEX:STX stack:LDX #7
2470 .init2 LDA addr,X:STA &38C,X
2480 STA pblk+2,X:LDA #0
2490 STA pblk+10,X:DEX:BPL init2:RTS
2500 :
2510 .addr OPT FNequd(&FFFF1900)
2520 OPT FNequd(&FFFF8023)
2530 .cmd OPT FNequs("*MERGE ")
2540 OPT FNequb(0)
2550 .cmd2 OPT FNequs("*PSAVE ")
2560 OPT FNequb(0)
2570 ]:NEXT
2580 ENDPROC
2590 :
2600 DEF FNequb(byte)
2610 ?P%=byte:P%=P%+1:=opt
2620 :
2630 DEF FNequd(num)
2640 !P%=num:P%=P%+4:=opt
2650 :
2660 DEF FNequs(A$)
2670 $P%=A$:P%=P%+LEN(A$):=opt
2680 :
2690 DEF PROCchecksum
2700 S%=0
2710 FOR J%=code TO P%-1:S%=S%+?J%
2720 NEXT:ENDPROC
2730 :
2740 REM Basic-I entry points and check
sum value
2750 DATA &BCAA,&BE88,&9B03
2760 DATA &8A13,&8A96,59562
2770 REM Basic-II entry points and che
cksum value
2780 DATA &BC8D,&BE6F,&9B29
2790 DATA &8A8C,&8AF3,59760
2800 REM Basic-IV entry points and chec
ksum value
2810 DATA &BB15,&BDE5,&9D3B
2820 DATA &8ED5,&8F83,59746
2830 REM Basic-VI entry points and chec
ksum value
2840 DATA &BA65,&BD45,&9D1B
2850 DATA &8EAE,&8F64,59563

```

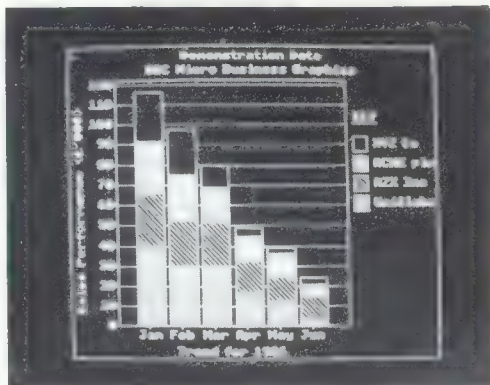
BUSINESS GRAPHICS (Part 1)

Graphs, charts and diagrams are the heart of any business data system. Alan Dickinson has pushed the Beeb to its limits to produce an outstanding business graphics system combining both power and flexibility with an easy-to-use approach.

The BBC micro has always received praise for its graphics capabilities, and Acorn provide some very powerful graphics routines as standard operating system software. We have enjoyed some quite breathtaking games, but hard-nosed business applications have been few and far between.

1986 was predicted to be the year of business graphics, (but so was '85, and '84). Windows and icons have become fashionable, but still we have seen relatively little business software to exploit these features. Here then is the BEEBUG Business Graphics package, a no-compromise program that pushes BBC Basic and the model B to its limits. By writing it in Basic, it will be easier for owners of shadow RAM cards, graphic extension ROMs, and Masters to expand the program and produce even more stunning displays, or to interface this program to the data files of other software.

This article, and program, will be in three parts. This month we present the basic graphics software, which will produce high quality bar charts and line charts on screen from data contained in DATA statements. Next month we will be printing the code for a menu-driven front end, allowing you easily to set up and modify graphs, and save the data independently of the program, and in the Jan/Feb issue we will be printing various extra routines to display pie charts and 3D charts. It will be much easier to combine these various bits of code if you keep to the line numbers used here.



USING THE PROGRAM

To see just how good the graphics are type in this month's listing and save to disc or cassette. Just run the program and it will immediately start to create a screen display of a bar chart complete with scales, title, legends etc. All the information to control the screen display, and the data used, is contained in DATA statements from line 5000 onwards, though this will change when part two is added.

By changing the values listed in the DATA statements you can vary all the parameters (and data) to see the range of graphs and charts that are possible. This is described in detail in the next section. Each time you modify this information you will then need to re-run the program to generate a new display.

UNDERSTANDING THE DATA

The first four pieces of data in the program are the title, subtitle, x-axis title and y-axis title. There is then a numeric value to describe the choice of chart layout with values of:

- 0 - Stacked bar chart
- 1 - Clustered bar chart
- 2 - Bar chart of first "ON" variable
- 3 - Range chart
- 4 - 3D chart
- 5 - Line chart
- 6 - Pie chart of first "ON" column

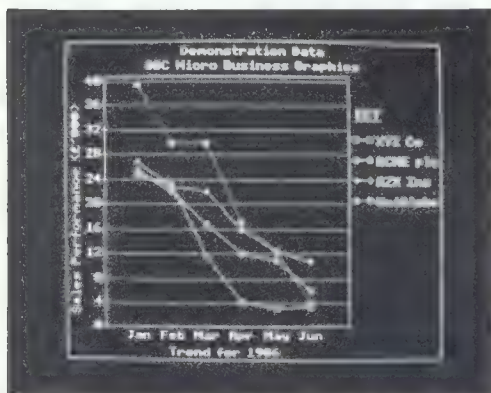
Options 4 and 6 will be implemented later, but it is well worthwhile re-running the program using each of the other display options in turn. The next value in the data is the grid flag, and when this is 1 a background grid will be drawn behind the bars (put a '0' if this is not needed).

Business charts frequently have time intervals along the x-axis of a bar chart, and up to 13 such intervals may be displayed. The next 4 lines of data describe a legend for each of the 13 columns, and a switch to indicate if each column is OFF (0) or ON (1).

Each column in the chart may have up to 7 pieces of data associated with it, (variables), and the next 4 lines of data describe the legend of each variable, a flag to indicate if the variable is ON or OFF, and a patterning code in the range 0-255 (used to shade the appropriate part of each chart).

The switches on the columns and the variables allow chunks of data to be switched off without actually being deleted, very useful if you just want to see what the graph looks like without one of the variables or one of the columns. This technique can also be used to display fewer than the maximum 13 columns and/or 7 variables as shown in the example data.

The final lines of data statements are the actual data, one line for each of the seven variables with one value (default 0) in it for each of the thirteen columns.



In operation, the program takes care of all aspects of graph layout. Titles are centred and text is proportionally spaced. Areas for legends are calculated from the lengths of the legends to ensure that maximum room is available for the plotting of the graph whilst retaining legibility of text, and scaling is carried out to ensure that the bars produced neatly fill

the plotting area. Every effort has been made to produce charts of the highest quality possible on a BBC screen.

```

10 REM Program GPROG (Part 1)
20 REM Version B1.1
30 REM Author Alan Dickinson
40 REM BEEBUG November 1986
50 REM Program subject to copyright
60 :
100 MODE4:VDU23,1,0;0;0;0;
110 VDU23,240,&88F8;&8888;&F8;0;23,241
,&D870;&D888;&70;0;23,242,&7020;&70F8;&2
0;0;23,243,&F8F8;&F8F8;&F8;0;
120 DIMC$(12),C$(12),V$(6),V$(6),H$(6)
,N(12,6):p$=STRING$(20," ")
140 PROCload:PROCdis:A%=GET
200 MODE7:END
210 :
1710 DEFPROCload
1720 READ $&980,$&9A0,$&9C0,$&9E0,Z%,C%
1730 FORJ%=0TO12:READ C$(J%),C%(J%):NEXT
T:FORJ%=0TO6:READ V$(J%),V%(J%),H%(J%):N
EXT
1740 FORJ%=0TO6:FORK%=0TO12:READ N(K%,J
%):NEXT:NEXT
1750 ENDPROC
1970 DEFPROCdis:VDU26
1975 IFZ%=40RZ%=6 PRINT""Option not av
ailable.""Press any key":ENDPROC
1980 PROCbox(0,0,1279,1023,0):PROCsetcv
1990 IFV%=0ORC%=0PROCzw(36,"NO DATA"):P
RINT""All variables and/or columns are
OFF""Press any key":A%=GET:ENDPROC
2000 PROCTitles:R%=1200:P%=72:IF(Z%<2)
AND(Z%<3)AND(Z%<6)AND(V%>1)PROClegend
2010 IFZ%<6PROCcyscal:ELSEPROCcolmax(D%)
2020 PROCxaxis:IFZ%<5PROCbarch:ELSEIFZ%
<6PROClinech:ELSEPROCpie
2030 ENDPROC
2040 DEFPROCsetcv:LOCALJ%:V%=0:W%=0:C%=
0:W%=0
2050 FORJ%=6TO0STEP-1:IFV%(J%)>0V%=V%+1
:W%=J%
2060 NEXT
2070 FORJ%=12TO0STEP-1:IFC%(J%)>0C%=C%+
1:D%=J%
2080 NEXT:ENDPROC
2090 DEFPROCTitles:Q%=112:S%=1003
2100 IF$&980<>""PROCTxt($&980,5,640,S%)
:S%=S%-48
2110 IF$&9A0<>""PROCTxt($&9A0,3,640,S%)
:S%=S%-48
2120 IFZ%<6PROCTxt($&9E0,11,20,511)
2130 S%=S%-16:ENDPROC
2140 DEFPROClegend
2150 K%=0:dK%=0:FORJ%=0TO6
2160 IFV%(J%)>0L%=FNTwid(V$(J%)):IFZ%=4
L%=L%-dK%:dK%=dK%+48

```

```

2170 IFL%>K%K%=L%
2180 NEXT
2190 R%=R%-K%:Y%=S%-96:IF(Z%<>3)AND(Z%<
>4)PROCTxt("KEY",83,R%,Y%):Y%=Y%-80:FORJ
%=0TO6:PROC1gbox(J%):NEXT:R%=R%-40:ENDPR
OC
2200 R%=R%-48*V%:S%=S%-48*V%
2210 X%=R%+48:Y%=Q%+24:FORJ%=0TO6
2220 IFV%(J%)>0PROCTxt(V$(J%),17,X%,Y%)
:X%=X%+48:Y%=Y%+48
2230 NEXT:ENDPROC
2240 DEFPROC1gbox(J%):LOCALh%:h%=H%(J%)
2250 IFV%(J%)=0ENDPROC
2260 IFZ%<4PROCbox(R%,Y%-40,R%+48,Y%+8,
h%)ELSEIFZ%=5PROCline(R%,Y%-8,R%+48,Y%-8
,h%)
2270 PROCTxt(V$(J%),19,R%+68,Y%)
2280 Y%=Y%-64:ENDPROC
2290 DEFPROCyscal
2300 PROCgmax:PROCscamax
2310 PROCdrscal:ENDPROC
2320 DEFPROCgmax:LOCALJ%
2330 gmax=0:gmin=999999999:gsum=0
2340 FORJ%=0TO12
2350 IFC%(J%)>0PROCcolmax(J%):PROCcolad
2360 NEXT:M%=gmax+.99:m%=gmin-.99:gave=
gsum/(V%*C%):IFm%>0m%=0
2370 ENDPROC
2380 DEFPROCcolad
2390 IFZ%=0 cmax=csum:cmin=0
2400 IFcmax>gmax gmax=cmax
2410 IFcmin<gmin gmin=cmin
2420 gsum=gsum+csum:ENDPROC
2430 DEFPROCcolmax(J%)
2440 cmax=0:cmin=999999999:csum=0
2450 IFZ%=2cmax=N(J%,W%):cmin=cmax:csum
=cmax:cave=cmax:ENDPROC
2460 FORK%=0TO6:IFV%(K%)>0 PROCcell
2470 NEXT:cave=csum/V%:ENDPROC
2480 DEFPROCcell:d=N(J%,K%)
2490 IFd>cmax cmax=d
2500 IFd<cmin cmin=d
2510 csum=csum+d:ENDPROC
2520 DEFPROCscamax:N%=M%:n%=m%:j%=1
2530 REPEAT:P%=P%+32:j%=j%*10:N%=N%DIV1
0:n%=n%DIV10:UNTILN%=0ANDn%=0
2540 j%=j%DIV100:IFj%=0j%=1
2550 M%=(M%+j%-1)DIVj%*j%:m%=(m%-j%+1)D
IVj%*j%:N%=M%:n%=m%
2560 IFN%>ABS(n%)T%=N%ELSET%=-n%
2570 t%=16*(T%/(N%-n%)):T%=T%DIVj%
2580 IFT%<t%REPEAT:T%=T%/2+.99:j%=j%*2
:UNTILT%<t%
2590 M%=(M%+j%-1)DIVj%*j%:m%=(m%-j%+1)D
IVj%*j%:Y%=(S%-Q%)/(M%-m%):IFm%<0P%=P%+32
2600 IFZ%=4 PROCcube:ELSEPROCbox(P%,Q%,
R%+16,S%+16,0)
2610 ENDPROC
2640 DEFPROCdrscal:PROCdrscal2(M%)

```

```

2650 IFm%=0ENDPROC
2660 MOVEP%,Q%-m%*Y:PLOT1,R%-P%+16,0:PL
OT0,0,-4:PLOT1,-(R%-P%+24),0
2670 PROCdrscal2(m%):ENDPROC
2680 DEFPROCdrscal2(a%)
2690 FORJ%=0TOa%STEPj%*SGN(a%)
2700 MOVEP%,FNypos(J%)
2710 PLOT1,-8,0:PLOT0,-4,16
2720 PROCTxt(STR$(J%),163,0,0)
2730 IF(G%=1)AND(Z%<>4)MOVEP%,FNypos(J%
):PLOT25,R%+16-P%,0
2740 IF(G%=1)AND(Z%=4)MOVEP%,FNypos(J%)
:PLOT1,s%,s%:PLOT25,R%-P%,0
2750 NEXT
2760 ENDPROC
2770 DEFFNypos(y)=Q%+(y-m%)*Y
2780 DEFPROCxaxis

```

PROGRAM NOTES

The program is quite tight on memory, and uses mode 4 as good resolution was felt to be more important than colour. Users with shadow RAM could of course have the luxury of both. The following routines will be of particular interest to readers converting the programs to use shadow RAM and mode 1.

PROCTxt(T\$,p,x,y) - line 3450

This routine writes proportionally spaced text T\$ at co-ordinates (x,y). The text can be centred over the point or aligned to start or end at that point. It can be rotated through 90 degrees for axis labelling, all under the control of parameter p. Most importantly though, the bottom two least significant bits of p select the colour of the text.

PROCbox(x1,y1,x2,y2,p) - line 3790

This routine draws a box on the screen at the co-ordinates (x1,y1)-(x2,y2). The parameter p is a value in the range 0-255 indicating how the box should be filled. The code in this routine is designed to give a good range of patterns in mode 4, but could easily be re-written to make use of either the four colours available in mode 1, or the pattern-fill capabilities of the Graphics Extension ROM (or Master Series), or even both.

PROCline(x1,y1,x2,y2,p) - line 3980

Likewise, this routine draws lines between the co-ordinates in either dotted or solid lines and with various point markers all under the control of the parameter p.


```

2790 IFZ%=6PROctxt($&9E0+" - "+C$(D%),3
,(P%+R%)/2,40):ENDPROC
2800 IF(Z%>2)AND(V%>1)PROctxt($&9C0,3,
(P%+R%)/2,40)ELSEPROctxt($&9C0+" - "+V$(
W%),3,(P%+R%)/2,40)
2810 PROClabcols:ENDPROC
2820 DEFPROClabcols:n%=1:X=(R%-P%)/(C%+
1):FORj%=0TO12
2830 :IFC%(j%)>0MOVEP%+X*n%,Q%:PLOT1,0,-
8:PLOT0,0,-8:PROctxt(C$(j%),131,0,0):n%=
n%+1
2840 NEXT:ENDPROC
2850 DEFPROCbarch:n%=0
2860 FORJ%=0TO12
2870 IFZ%=0IFC%(J%)>0PROCstack(n%)
2880 IFZ%=1IFC%(J%)>0PROCclust(n%)
2890 IFZ%=2IFC%(J%)>0PROCbar(n%)
2900 IFZ%=3IFC%(J%)>0PROCrange(n%)
2910 IFZ%=4IFC%(J%)>0PROCthreed(n%)
2920 IFC%(J%)>0n%=n%+1
2930 NEXT:ENDPROC
2940 DEFPROCstack(n%):LOCALA%,B%,X%,Y%
2950 A%=P%+X*n%+16+X/2:B%=Q%:X%=A%+X-16
2960 FORK%=6TO0STEP-1
2970 IFV%(K%)>0D=N(J%,K%):Y%=D*Y+B%:PRO
Cbox(A%,B%,X%,Y%,H%(K%)):B%=Y%
2980 NEXT:ENDPROC
2990 DEFPROCclust(n%):LOCALA%,B%,X%,Y%
3000 A%=P%+X*n%+X/2+8:X%=A%+X-32
3010 skip=(X%-A%)/V%
3020 FORK%=0TO6
3030 IFV%(K%)>0Y%=FNypos(N(J%,K%)):PROC
box(A%,FNypos(0),A%+skip,Y%,H%(K%)):A%=A
%+skip
3040 NEXT:ENDPROC
3100 DEFPROCbar(n%):LOCALA%,B%,X%,Y%
3110 A%=P%+X*n%+X/2:B%=FNypos(0)
3120 X%=A%+X:Y%=FNypos(N(J%,W%))
3130 PROCbox(A%,B%,X%,Y%,H%(W%))
3140 GCOL0,0:MOVEX%+4,Y%-4:DRAWX%+4,B%+
4
3150 GCOL0,1:MOVEX%,Y%:DRAWX%+8,Y%-8:DR
AWX%+8,B%
3160 IFN(J%,W%)<0MOVEX%+8,Y%-8:PLOT1,-X
,0:PLOT1,8,8
3170 ENDPROC
3180 DEFPROCrange(n%):LOCALA%,B%,X%,Y%:
PROCcolmax(J%):VDU5:A%=P%+X*n%+X:B%=FNyp
os(cmin):Y%=FNypos(cmax)
3190 MOVEA%-8,B%+8:VDU243:MOVEA%-8,Y%+8
:VDU243:MOVEA%-4,B%:DRAWA%-4,Y%:MOVEA%+4
,B%:DRAWA%+4,Y%:MOVEA%-8,FNypos(cave)+8:
VDU243:VDU4:ENDPROC
3200 DEFPROcline:FORJ%=0TO6
3210 IFV%(J%)>0PROCtrace(J%)
3220 NEXT:ENDPROC
3230 DEFPROCtrace(J%):LOCALN%,K%,L%
3240 N%=0:L%=0:FORK%=0TO12
3250 IFC%(K%)>0IFN%>0PROcline(P%+X*N%,F
Nypos(N(L%,J%)),P%+X*(N%+1),FNypos(N(K%,
J%)),H%(J%))
3260 IFC%(K%)>0L%=K%:N%=N%+1
3270 NEXT:ENDPROC
3450 DEFPROCtxt(T$,P%,X%,Y%):LOCALC%,Q%
,E%,W%,J%:IFT$=""ENDPROC
3460 E%=1:IF(P%AND8)=0Q%=0ELSEQ%=1
3470 IF(P%AND16)>0THENE%=0
3480 IF(P%AND32)>0THENE%=2
3490 GCOL0,(P%AND3):IF(P%AND128)=0MOVEX
%,Y%ELSEPLOT0,X%,Y%
3500 W%=Fntwid(T$)
3510 IFQ%=0ANDE%=1PLOT0,-W%DIV2,0
3520 IFQ%=0ANDE%=2PLOT0,-W%,0
3530 IFQ%=1ANDE%=1PLOT0,0,-W%DIV2
3540 IFQ%=1ANDE%=2PLOT0,0,-W%
3550 IF(P%AND64)>0ANDQ%=0PLOT0,0,-28:PL
OT1,W%,0:PLOT0,-W%,36
3560 IF(P%AND64)>0ANDQ%=1PLOT0,36,-32:P
LOT1,0,W%:PLOT0,-36,-W%+32
3570 VDU5:FORJ%=1TOLEN(T$)
3580 C%=ASC(MID$(T$,J%,1)):W%=Fncwid(C%
)
3590 IFQ%=1PROCinv:C%=228
3600 VDUc:IFQ%=0PLOT0,W%-32,0 ELSEPLOT
0,-32,W%
3610 NEXT:VDU4:ENDPROC
3620 DEFFntwid(T$):LOCALJ%,K%,C%
3630 FORJ%=1TOLEN(T$)
3640 T%=T%+Fncwid(ASC(MID$(T$,J%,1)))
3650 NEXT:T%
3660 DEFFncwid(C%):LOCALX%,Y%,A%,j%,k%
3670 X%=&70:Y%=0:A%=&A:?X%=C%:CALL&FFF1
3680 j%=?&71OR?&72OR?&73OR?&74OR?&75OR?
&76OR?&77OR&78
3690 IFj%=0THEN=16
3700 k%=36:j%=j%*2:REPEAT
3710 j%=j%DIV2:k%=k%-4
3720 UNTIL(j%AND1)<0:=k%
3730 DEFPROCinv:LOCALJ%,K%,L%
3740 VDU23,228:FORJ%=1TO8:L%=0
3750 FORK%=&71TO&78:L%=L%*2
3760 IF(?K%MOD2)=1L%=L%+1
3770 ?K%=?K%DIV2:NEXT:VDUL%
3780 NEXT:J%=Fncwid(228):ENDPROC
3790 DEFPROCbox(P%,Q%,A%,B%,C%)
3800 LOCALI%,J%,K%,L%,M%,N%,S%,X%,Y%
3810 IFP%>A%I%=A%:A%=P%:P%=I%
3820 IFQ%>B%I%=B%:B%=Q%:Q%=I%
3830 GCOL0,1:MOVEP%,Q%:DRAWP%,B%:DRAWA%
,B%:DRAWA%,Q%:DRAWP%,Q%
3840 IFA%-P%<12ORB%-Q%<12ENDPROC
3850 VDU24,P%+4,Q%+4,A%-4,B%-4;
3860 GCOL0,128+(C%AND1):CLG
3870 IFC%<2VDU26:ENDPROC
3880 GCOL4,0:S%=8:IF(C%AND64)>0S%=12
3890 I%=P%DIV16*16:K%=(A%+15)DIV16*16
3900 J%=Q%DIV16*16:L%=(B%+15)DIV16*16
3910 Y%=L%-J%:X%=K%-I%

```

MICRO TRADER

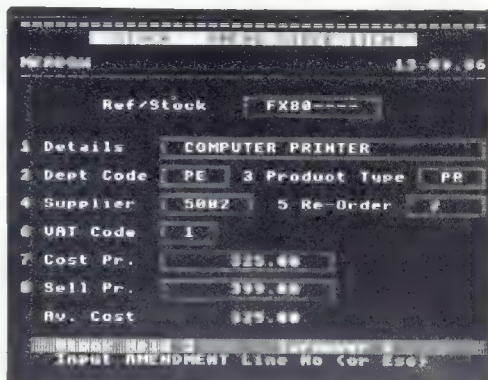
Micro Trader, an integrated accounts package for the small business, has been quietly building a reputation for itself in recent months. Simon Williams has been assessing what it has to offer the business user.

Product : Micro-Trader
Supplier : Meadow Computers
11 London Street,
Whitchurch, Hants RG28 7LH.
Tel. (025682) 2008
Price : £230.00 plus £110.00 for the
stock module (Micro-Stock).
Price includes 4 discs and
A4 manual.

If you talk to anyone running their own business, they'll nearly always admit that they spend more time doing their accounts and preparing VAT returns than any other form of administration. Hardly surprising then that accounting software has proved so popular among the uses to which the BBC Micro has been put. To be of real use, though, accounting software has to be reliable and easy to use. Combining these two requirements with the flexibility to tailor the package to many different businesses is quite a task.

Micro Trader was written by a businessman with all these factors in mind, and is straightforward and robust without becoming bogged-down with 'help' messages. The basic package is supplied as four discs and a 60 page A4 manual which is daisy-wheeled rather than being printed. This doesn't detract from the presentation, which is clear and legible, nor the content, which is concise and informative without being patronising or full of jargon.

The version supplied for review contained the optional stock control module, which is fully integrated with the rest of the system. It is also available separately though, or as a supplement to the main package which consists of the three Sales, Purchase and Nominal ledgers.



The idea of integrating the stock control module is that entering details of sales or purchases in the ledgers automatically updates the stock levels, removing the need to enter the same transaction twice.

SETTING UP

When you run Micro-Trader you are first presented with a sign-on screen. The whole of Micro-Trader uses mode 7 screens, which limits the amount of information which can be displayed at any one time, but does allow coloured highlights. You have to enter the date on this screen, but no password. There is no sign-on protection, although you have to enter a security code to use the final accounts module. This is a reasonable approach as Micro-Trader is largely aimed at the small (often one-person) business.

After entering the name and address of your company, which is used automatically to head invoices and statements, You can start to fill up your stock list, add Nominal ledger accounts (although there is a wide variety of pre-defined accounts), and enter your purchase and sales account details. Each sales account may be assigned a discount level.

SALES LEDGER

This ledger can hold up to 450 accounts for regular customers. Transactions can be fed into any of these accounts as goods are sold to them. Invoices are issued by the system and may be printed out on any Epson-compatible printer.

If the customer doesn't pay promptly (and many don't these days!), you can

Statement					
MEADOW COMPUTERS					
R. J. SMITH + Co Long Lane Westbury, Avon BS20 8RR			11, London Street Whitchurch Hants RG28 7LH TEL: (025682)2008 V.A.T. No 362 9561 31		
Account No.		Statement No.		Statement Date	
1003		13.09.86		8/86	
Value	Date	Invoice No.	T/X	Goods Value	VAT
	Credits	Invoice Value			
Brought Forward					
17.08.86	123 IN	450.00	64.13		
17.08.86	456 CH			2159.68	
17.08.86	456 SD			55.38	
STATEMENT TOTALS		450.00	64.13	2215.06	
SETTLEMENT DISCOUNT DEDUCTIBLE AT 5 %					
£ 22.50 If Paid By 30 of Month					
Aged Balances		1 MONTH	2 MONTHS	3 MONTHS++	
		0.00	514.13	0.00	
Remittance Advice					
MEADOW COMPUTERS					
R. J. SMITH + Co			Statement Date 13.09.86		
Account No 1003			Balance Due 514.13		
SETTLEMENT DISCOUNT DEDUCTIBLE AT 5 %					
£ 22.50 If Paid By 30 of Month Remittance Enclosed £					

issue statements individually or bracketed by lateness! These statements look suitably 'official' and include a pre-printed remittance advice for the tardy customer to tear off and return with his or her payment.

Sales reports include an Aged Debtors report (not a list of all your customers over 70!) and a bulk print of all outstanding statements. This report excludes any accounts with a zero or positive balance, which saves your postage and your customers' tempers.

PURCHASE LEDGER

This ledger has the same size limits as the sales ledger and handles all purchases of goods or services into the company, whether from one of your regular suppliers or on a one-off basis.

Each purchase can be posted to the appropriate Nominal ledger account. The system maintains a Journal so that you can make arbitrary postings to correct mistakes.

A number of reports are available from the Purchase ledger including an audit trail and an account report, and as with reports from the Sales ledger, these can be directed to the screen or the printer. No 'Suggested Payments' report is available to help you hang on to your money as long as possible, although some of the information can be gleaned from the 'Aged Creditors' report.

NOMINAL LEDGER

This is where you assign all the money going into and out of your company. The Nominal ledger contains up to 400 named accounts and can handle up to 4000 transactions a month.

Forty-odd pre-defined accounts cover the main areas of business activity,

but you can add to these to customise the ledger to your own needs. Each account belongs to one of 20 pre-defined 'types' which are used by Micro-Trader to link together groups of accounts.

When you come to post your transactions to particular Nominal ledger accounts, the program uses three function keys to select options from menus. Each key controls a selection from one of three menus and each is a refinement of the final choice. Overall this is a quick and convenient way of working.

STOCK CONTROL

The stock control module offers a database of up to 4000 items and holds details of price, stock code, description, product type and supplier. You can hold up to 9999 of any item and turnover up to 99999 in a year.

Each stock record is of a fixed format but there is a nice touch in the calculation of the average cost price. If the price of the item changes, so you have two or more batches of the same product in stock at different prices, the average cost is maintained by the program and used when valuing stock.

When you raise an invoice, the details of any stock items are automatically fetched from this module and inserted into the sales documentation.

When you come to the end of an accounting period (and Micro-Trader allows you to define several of these) all end-of-period calculations are done by the computer, which then closes off the accounts so you can't go back and make alterations.

Before it does this, though, it produces a report for auditing the system and a nominal ledger listing. At the end of the financial year the year-end final accounts routine will produce all the necessary documentation and then clear your books for the start of the new one. It will also produce a VAT report.

Micro Trader handles VAT within your accounts and can produce all your VAT returns in a suitable form for the tax officer. You can define up to ten different VAT rates.

When you close the package down at the end of the day, you're guided through a backing-up sequence which tries to ensure that you're never left without data, even if two out of three sets of discs are destroyed. The manual includes a 'data logging' sheet to help you keep track of the rotation of your disc sets. This is typical of the attention to detail throughout Micro-Trader.

One of the very effective features of Micro-Trader is the management information

***** *STOCK LEDGER-STOCK LIST* *****									
MEADOW COMPUTERS								DATE 13.09.86	
Ref	Stock	Detail	Unit	Prod.	Quantity	Cost	Sell		
1195		LISTING PAPER BOX 1000	ME	PA	9	1.50	6.50		
1248P		PRINTER RIBBON CARTRIDGE	ME	RR	0	2.95	3.50		
1251		COLOUR MONITOR	FE	VC	2	135.00	199.00		
1301		GREEN SCREEN MONITOR	FE	VC	5	59.00	75.00		
AB1123		64K MICROCOMPUTER	ME	VC	2	275.00	345.00		
10045		SINGLE DISC DRIVE 100K	FE	DI	1	75.50	110.00		
10080		DUAL DISC DRIVE 800K	FE	DI	1	210.00	275.00		
11000		BOX 10 DISCS - S/S S/D	ME	FE	6	9.75	11.50		
11010		BOX 10 DISCS - D/S S/D	ME	FE	6	11.50	19.50		
11020		FLOPPY DISC - D/S D/D	ME	FE	245	1.80	2.30		
11030		COMPUTER PRINTER	FE	FR	3	325.00	399.00		
11040		RED TAPE (PER YARD)	ME	RT	91.00	1.25	0.97		
Report Totals						9850	15100.00		

it produces. You can select a wide variety of stock reports, including a re-order level listing, which shows which items have fallen below your pre-set re-order levels. Any of these stock reports can be tailored by selecting particular department codes, product types and supplier codes, or a combination of these. These selective reports are very useful, particularly when processing a set of stock orders.

CONCLUSION

As you may have gathered, I'm impressed with the Micro-Trader package. It seems to be well thought out and thorough, and compares very favourably with the Peartree Business System (reviewed Vol.5 No.3) or the older, and more expensive, Integrated Accounting Suite from Gemini. It's flexible and should be able to handle the accounts of quite large businesses, given a maximum turnover of £21 million, with up to £1 million in any single account.

You'll need twin 80 track drives to run with your BBC Micro or Master, and an Epson compatible printer. Anyone thinking of business applications for a micro should consider this as a minimum configuration, anyway.

Anyone considering the use of their BBC Micro for accounting in a small-to-medium sized business should certainly include Micro-Trader on their shortlist of possible packages.

DB

Multi-column Wordwise Plus

Ian Waugh describes how he has augmented Wordwise Plus to provide a multiple column printout facility, readily tailored to your own requirements.

INTRODUCTION

There are many occasions when a multicolumn printout of your text would be useful. Columns are easier to read and more attractive to the eye than a block of text. Instructions and game rules and even reports can benefit from a multi-column format.

The routine presented here attempts to overcome the restrictions imposed by the dual-column segment program included with the Wordwise Plus package. It can print text in 2 or 3 (or even more) columns using elite, pica, condensed or any other form of print. You can set the column widths and vary the spacing between columns. You can also see how a column is going to look on the page and so prevent such layout disasters as a heading on the bottom of a page with a blank line and the text at the top of the next column.

COMPATIBLE PRINTERS

The program was written for the Epson FX80 printer and the columns are produced by using its reverse feed facility. The routine should be compatible with, or easily adapted to any

printer with reverse feed. If your printer does not have this facility, you can pause the printing and rewind the sheet by hand by substituting a suitable routine in place of PROCTop in the program. Reverse feed is also used to print the headers and footers so you may have to forgo these unless you can think of a way around it. To set the column widths the program sends Epson codes directly to the printer with VDU1.

USING THE PROGRAM

Type the program into any convenient segment and save it before running it. When it is debugged you can remove the REMs and abbreviate the commands to save memory. It is recommended that you also keep a REMed version so you can easily refer to and alter those variables which determine the print format.

To prepare your text for printing, set the initial format at the start of your text. This article uses the following:

```
<f1>PL66<f1>TS3<f1>BS3<f1>L
L27<f1>JD<f1>EP<f1>DF
```

Headers and footers are defined in the program and should not be defined in the text. Their positions on the page can be altered by changing the amount of reverse feed in their print routines. The program inserts <f1>BP at the end of the text in order to print the last page correctly. If you run the routine more than once, don't forget to delete this or else write a small routine to do it for you.

The default values in the program will produce a page layout similar to this

one but you will want to alter the header. Switch on your printer, insert some paper and run the program by pressing Shift and the function key which corresponds to the segment the program is in.

ALTERING THE PRINT FORMAT

You can change the layout by altering three variables: E%, N% and S%. E% is used to calculate the column spacing and must be set to the number of characters per line. Print mode is selected by the program and should not be altered in the text although *italics* and underlining are okay. N% is the number of columns you require and S% is the number of spaces between the columns.

T% is the number of normal pages that will be produced from the columns. The variable P% is used by Wordwise Plus to hold a copy of the current page number. It is updated after previewing the text and used to calculate the value of T%. The section of code which does this will delete any markers in the text.

M% is the line length and L% is the page length. These are picked out of the text automatically from LL and PL respectively. The values given in the program are only suggestions but they produce well-spaced text.

The print positions of the columns, AX, BX and CX, are calculated before the printing starts. The first column is the most complex to calculate but after that the others simply add on the column line length (M%) and the column spacing (S%). It would therefore be

an easy matter to produce print in four, five or more columns.

HINTS

Before printing your text, preview it to check that the column breaks are neat. Line number start (LNS) can be very helpful in this respect to ensure perfectly aligned text. Finally, if reverse feed leaves a line on the paper, unlock the paper release lever.

```
REM Wordwise Plus Multi-Column Printer
REM Version 1.1
REM Copyright BEEBUG (C) November 1986
REM E% = number of characters per line
REM On standard 11"x9.5" listing paper
REM the following values apply:
REM Pica: E%=80
REM Elite: E%=96
REM Condensed: E%=137
E%=96
REM Select print mode. This is elite
VDU2,1,27,1,33,1,1,3
REM Number of columns. Set to 2 or 3
N%=3
REM Number of spaces between columns
S%=3
REM T% = number of 'normal' pages
SELECT TEXT
DELETE MARKERS
CURSOR BOTTOM
FKEY3
FKEY3
DEFAULTS
PREVIEW MARKED
DELETE MARKERS
T%=(P%+N%-1)/N%
REM Insert <f1>BP to print last page
TYPE " GBP"
REM Suggested line length (LL) in elite
REM For 2 columns = 35
REM For 3 columns = 29
REM Get LL from text. Set M%=LL
CURSOR TOP
FIND "LL"
CURSOR RIGHT 2
L$=GCT$
L$=L$+GCT$
M%=VALL$
REM L% = page length
REM Get PL from text. Set L%=PL
CURSOR TOP
FIND "PL"
CURSOR RIGHT 2
L$=GCT$
L$=L$+GCT$
L%=VALL$
```

```
REM Start print position of 1st column
A%=(E%-(M%*N%+S%*(N%-1)))/2
REM Start print position of 2nd column
B%=A%+M%+S%
REM Start print position of 3rd column
C%=B%+M%+S%
REM Define header
H$="Beebug 2/3-Column Demo "
SELECT TEXT
REM Page number in text
X%=1
DOTHIS
REM Left margin for 1st column
VDU2,1,27,1,108,1,A%,3
```

```
PROCprintpage
REM Define and print footer/page number
VDU2,1,27,1,108,1,E%/2-3
P$="PAGE "+STR$(X%-1) DIV N%+1)
PRINT P$
VDU1,27,1,106,1,36
PROCtop
REM Print header
REM Add P$ (ie page number) if required
D$=H$+P$
VDU2,1,27,1,108,1,E%-LEND$-4
PRINT D$
VDU1,10
VDU2,1,27,1,106,1,36
VDU1,27,1,106,1,36,3
REM Left margin for 2nd column
VDU2,1,27,1,108,1,B%,3
```

```
PROCprintpage
IF N%=2 THEN GOTO two
```

```
PROCtop
REM Left margin for 3rd column
VDU2,1,27,1,108,1,C%,3
```

```
PROCprintpage
.two
TIMES T%
```

END

```
.printpage
DEFAULTS
PRINT PAGE X%
X%=X%+1
ENDPROC
```

```
REM Back to top of page
.top
VDU2
DOTHIS
VDU1,27,1,106,1,36
TIMES L%
VDU3
ENDPROC
```




Beebug News

THE BEEBUG PAGE DESIGNER

PRINTER GRAPHICS
(Part 3)

Alex Kang completes his series on printer graphics with a full-blown page designer for Epson compatibles.



The Beebug Page Designer is a self-contained package which allows the user to design and print pages of mixed graphics and text in a wide variety of layouts. This means that elaborate magazine-style pages may be created, with illustrations appearing anywhere on the page. Alternatively the Page Designer is equally at home with logos and letterheads. The system is flexible and its uses are legion.

Page Designer contains all the routines that you will need for the task, including a windowed screen dump. All that you need to supply are suitable mode 4 screen dumps containing your designs and illustrations.

INSTALLING THE PAGE DESIGNER

Because of memory restrictions, the Page Designer is presented as two separate programs, with menu options to move between them. The two programs should be typed in separately and saved under the names DESIGNA and DESIGNB. The third program accompanying this article is shared by both DESIGNA and DESIGNB, and needs to be included with each. To avoid typing this part twice you can merge it with both parts, either by using the merge utility found elsewhere in this issue, or by proceeding as follows. Type in this shared program, and save it away. Then make a spooled copy of it by typing:

```
*SPOOL SPOOLED
LIST
*SPOOL
```

Now to append this spooled file to each of the other two parts, simply load in

DESIGNA, and type:

```
*EXEC SPOOLED
```

You now have a merged version of DESIGNA containing the shared lines. Save this away under the name DESIGNA, and repeat the process with DESIGNB.

Even with the programs split in this way however, PAGE must be set to &1300 or below on a machine without shadow memory (i.e. type PAGE=&1300 before loading the programs). To run the pair of programs, simply CHAIN the first, i.e. type:

```
CHAIN "DESIGNA"
```

USING THE PAGE DESIGNER

The program DESIGNA is concerned with the creation of new print formats, as we will call them. A format consists of a graphics window (unfilled) accompanied by text either to the right, left or on both sides of it. The window is not filled with a cropped screen dump until the printout stage. In making up a page of text one would first create a number of formats, and then combine them together at printout stage to make a complete page. Alternatively a single shallow format could be used to produce an embellished letterhead for example.

When DESIGNA is run it offers three menu options:

1. Load/View format
2. Create new format
3. Print

The first two options are fairly self-explanatory, with option 1 allowing star commands. Option 3 simply calls DESIGNB to initiate the screen selection and printing process.

Option 2 provides you with a fresh mode 0 design screen, representing the printer paper, so that what you format on this screen will be what you get on the printout. If you select 2, you must first use the cursor keys and space bar (as in FLEXIDUMP) to define the required graphics window. The scales at the top and bottom of the design area correspond to the eighty text columns available, and the maximum window size approximately equates to that of a full size mode 4 dump. Once the lower left hand corner of the window has been marked with the space bar, a new scale will appear representing the text co-ordinates of the mode 4 graphics area. This is to assist you in determining the size of the window.

Once the window has been defined, text may be entered, with Return taking you to the next line (or the next part of the line, if text is split across a window). Lines may be drawn using the underline character, but beware of leading and trailing spaces in the text: these are treated as valid text characters during subsequent justification. Once the cursor reaches the bottom of the design area, you may edit any line, after which you are asked to select left, right or central justification. The format should then be saved away. If you reply in the negative to the 'save' prompt the format will be cleared.

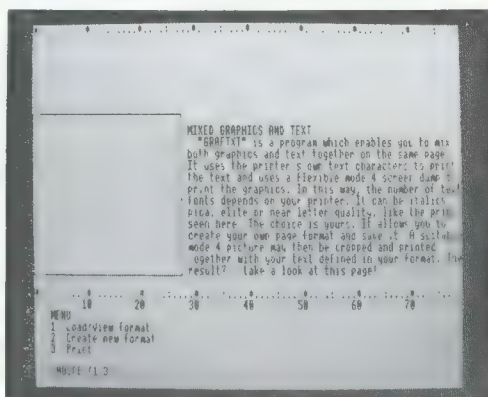
PAGE MAKE-UP

When the menu returns, selecting option 3 initiates the printout process, chaining DESIGNB. For each segment of text and graphics, you are asked to supply the filename of a previously saved format. Again star commands may be used, and *CAT will be particularly useful here. Next you must select a font style. As the program stands, you are offered Emphasized, Italics or NLQ, but if your printer does not have an NLQ facility, you could substitute some other font for this by replacing its code in PROCon and PROCOff in DESIGNB.

Once the font has been selected you are asked for the filename of a suitable mode 4 screen. This is loaded in and displayed, and a window the size of that defined in the format you have loaded will be overlaid. Use the cursor keys to position it, with Shift speeding up movement, and the space bar to freeze the position. Once the graphics window has been defined, you are offered the option of a boxed or unboxed window, and then printout commences. If you require more than a single segment of text and graphics, you may then select another format, and repeat the process until a page has been built up.

While experimenting with the Page Designer, you will probably think of all sorts of uses for it, like printing letterheads, cards, newsletters, or simply just to decorate your letters. Perhaps you may like to enhance the program's features by adding full justification, improving the text editing facilities, accommodating screens in other graphics modes and so on. It is also hoped that this short series

may have given you ideas for writing other printer applications.



TECHNICAL NOTES

From discussions last month you may recall that for normal printing on Epson printers, lines of text are 12 dots apart, but that in the graphics mode, each scan is only 8 dots high. We get around this by printing a line of text and graphics in sequence, then issuing a linefeed of 8/72 ins (equivalent to 4 dots in height), and then printing the next line of graphics only, with a height of 4 dots.

The program DESIGNB contains an integral mode 4 windowed screen dump with a different scaling to that of FLEXIDUMP featured last month. The present dump has a built-in reduction effect, giving a relatively high definition picture, appropriate to the size of a printed page. It uses a printer grid of 640x256 dots to represent the 1280x1024 mode 4 screen. A full mode 4 screen would thus occupy exactly 2/3 of the width of the printer paper.

```
10 REM Program DESIGNA
20 REM Version B 0.36
30 REM Author Alex Kang
40 REM Beebug November 1986
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO 5410
110 IF PAGE>1300 THEN MODE 7:PRINTTAB
(5,10)"Please reload program with PAGE":
PRINTTAB(5,12)"set to &1300":END
120 MODE0:xs%=16:ys%=32
130 DIM TS(2,21),SS(2),name 25
140 n$="123":y$=" (Y/N) ? ":ys$="YyNn"
```



```

150 VDU19,0,7;0;:VDU19,1,0;0;
160 REPEAT
170 REPEAT
180 q$="MENU":PROCwin:PRINT"1. Load/V
iew format""2. Create new format""3. P
rint""CHOICE (1-3): ";:REPEAT:AS=GET$:
UNTILINSTR(n$,AS):PRINTAS:CC%=VAL(AS)
190 PROCdlay(20):VDU26:IF CC%=1:PROCld
f:PROCdisf
200 UNTIL CC%>1
210 IF CC%=3:CHAIN "DESIGNB"
220 CLS:PROCdga:PROCcalc
230 *FX15,1
240 PROCtext:PROCedit
250 J%=FNqa("JUSTIFICATION (1-3):"+CHR
$10+CHR$13+"1) Left 2) Centre 3) Right
:",n$):PROCjt:IF FNqa("Save format"+y$,
ys$)=TRUE THEN PROCsvf ELSE VDU26,12
260 UNTIL 0
270 :
1000 DEFFNqa(q$,p$)
1010 PROCwin:REPEAT:AS=GET$:UNTILINSTR(
p$,AS)>0:PRINTAS:VDU26
1020 IF AS$="Y" OR AS$="y":=TRUE
1030 IF AS$="N" OR AS$="n":=FALSE
1040 =VAL(AS)
1050 :
1060 DEFPROCwin:VDU28,3,31,79,25,12:PRI
NTq$::ENDPROC
1070 :
1080 DEFFNip(N%)
1090 J%=0:q$="":REPEAT
1100 A%=GET:IF J%=0 AND A%=127 VDU7:GOT
O1100
1110 IF A%=127 J%=J%-1:PRINTCHRSA%::q$=
LEFT$(q$,J%)
1120 IF J%<N% AND A%<13 AND A%<127 q$
=q$+CHRSA%:J%=J%+1:PRINTCHRSA%;
1130 IF J%=N% AND A%<13 VDU7
1140 UNTILA%=13
1150 =q$
1160 :
1170 DEFPROCdga
1180 PROCrul:PROCt("Use cursor keys and
<SPACE> to mark left limit of graphics
area."):X%=216:Y%=320
1190 REPEAT:PROCmove(0,1248,320,320)
1200 PROCbox(X%,Y%,X%+847,Y%+671,6):PRO
Cbox(X%,Y%,X%+847,Y%+671,6)
1210 UNTILsp$:VDU7:PROCdlay(20):*FX15,1
1220 P%=X%:X%=X%+847:Q%=Y%:Y%=Y%+671
1230 PROCt("Use cursor keys and <SPACE>
to mark top right corner of graphics ar
ea."):PROCvrl:REPEAT:PROCmove(P%+31,P%+
847,Q%+31,Q%+671):PROCbox(P%,Q%,X%,Y%,6)
:PROCbox(P%,Q%,X%,Y%,6)
1240 UNTILsp$:VDU7:PROCvrl:PROCbox(P%,
Q%,X%,Y%,6):IF X%>1279 X%=1279
1250 U%=X%:V%=Y%
1260 ENDPROC

```

```

1270 :
1280 DEFPROCcalc
1290 B%=P%/16:E%=80-(U%+1)/16:D%=6*(U%
-P%+1)/16:H%=12*((V%-Q%+1)/32):T%=H%/12
:EL%= (1023-V%)DIV32:S$(1)=STRING$(B%," "
):S$(2)=STRING$(E%," ") :ENDPROC
1300 :
1310 DEFPROCvrl
1320 VDU5:GCOL3,1:MOVEP%,320:DRAW1279,3
20:i%=P%:c%=0:REPEAT:c%=c%+1:i%=i%+64:MO
VEi%-8,332:PRINT;"+":MOVEi%-8,360:PRINT;
c%*3-1:UNTILi%>=1279 OR c%*3-1=38
1330 VDU5:GCOL3,1:MOVEP%,320:DRAWP%,960
:FORi%=1TO10:MOVEP%,320+i%*64:PRINT"-";i
%*3:NEXT:VDU4:ENDPROC
1340 :
1350 DEFPROCrul:q$="*.....":PRINTTA
B(0,0)STRING$(8,q$);TAB(0,23)STRING$(8,q
$):FORi%=1TO7:PRINTTAB(i%*10-1,24);i%*10
:NEXT:ENDPROC
1360 :
1370 DEFPROCtext
1380 PROCt("Type text lines now."):FORi
%>TOT%-1:IF B%<0 VDU31,0,EL%+i%:T$(1,i
%)=FNem(FNip(B%),B%)
1390 IF E%<0 VDU31,80-E%,EL%+i%:T$(2,i
%)=FNem(FNip(E%),E%)
1400 NEXT:ENDPROC
1410 :
1420 DEFPROCjt:FORi%>TOT%-1:IF B%>0 GO
TO1450
1430 IF J%=2 THEN T$(1,i%)=FNcn(T$(1,i%
),B%) ELSE T$(1,i%)=FNlrrj(T$(1,i%),B%)
1440 PRINTTAB(0,EL%+i%);T$(1,i%)
1450 IF E%>0 GOTO1480
1460 IF J%=2 THEN T$(2,i%)=FNcn(T$(2,i%
),E%) ELSE T$(2,i%)=FNlrrj(T$(2,i%),E%)
1470 PRINTTAB(80-E%,EL%+i%);T$(2,i%)
1480 NEXT:ENDPROC
1490 :
1500 DEFFNcn(q$,Z%):LOCAL s%,s1%
1510 IF (Z%-LENq$)<2:=q$
1520 s%=(Z%-LENq$)DIV2:s1%=(Z%-LENq$)MO
D2:IF s1%=0:=STRING$(s%," ") +q$+STRING$(
s%," ")
1530 =STRING$(s%,CHR$32)+q$+STRING$(s%+
1,CHR$32)
1540 :
1550 DEFFNlrrj(q$,Z%)
1560 IF LENq$=Z%:=q$
1570 IF J%=1:=q$+STRING$(Z%-LENq$," ")
1580 IF J%=3:=STRING$(Z%-LENq$," ") +q$
1590 :
1600 DEFPROCedit
1610 IF FNqa("Edit text"+y$,ys$)=FALSE:
ENDPROC
1620 i%=0:ty%=- (80-E%)*(B%=0):PROCt("Se
lect line using cursor keys and spacebar
"):REPEAT:IF INKEY-58 AND i%>0:i%=i%-1
1630 IF INKEY-42 AND i%<T%-1:i%=i%+1

```

```

1640 IF INKEY-122 AND E%>0 ty%=80-E%
1650 IF INKEY-26 AND B%>0 ty%=0
1660 VDU31,ty%,EL%+i%:PROCdlay(10):*FX1
5,1
1670 UNTILINKEY-99:VDU7:*FX15,1
1680 PROCT("Type in correct text"):VDU3
1,ty%,EL%+i%:LL%=2+1*(ty%=0):ip%=-B%*(LL
%=1)-E%*(LL%=2):T$(LL%,i%)=FNip(ip%):IF
FNqa("Edit again"+y$,ys$)=TRUE GOTO1620
1690 ENDPROC
1700 :
1710 DEFPROCsvf
1720 q$="Save format filename. ":PROCwi
n:INPUT p$:VDU26:C%=OPENUPp$:IF C%>0 VDU
7:PROCT("File exists!"):PROCdlay(50):CLO
SE#C%:IF FNqa("Overwrite file"+y$,ys$)=F
ALSE GOTO1720
1730 C%=OPENOUTp$:PRINT#C%,D%,H%,T%,B%,
E%:FORi%=1TO2:PRINT#C%,S$(i%):NEXT:FORi%
=0TOT%-1:PRINT#C%,T$(1,i%),T$(2,i%):NEXT
:CLOSE#C%:PROCT("Format saved."):PROCdla
y(50):ENDPROC
1740 :
1750 DEFPROCdisf:PROCru1
1760 V%=319+(32*H%/12):EL%=(1023-V%)DIV
32:PROCbox(B%*16,320,16*(80-E%)-1,V%,5):
FORi%=0TOT%-1:IF B%<>0:VDU31,0,EL%+i%:PR
INTT$(1,i%)
1770 IF E%<>0:VDU31,80-E%,EL%+i%:PRINTT
$(2,i%)
1780 NEXT:ENDPROC
1790 :
1800 DEFFNem(p$,A%):IF p$="" THEN =STRI
NG$(A%," ")
1810 =p$

```

```

10 REM Program DESIGNB
20 REM Version B 0.35
30 REM Author Alex Kang
40 REM Beebug November 1986
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO 5410
110 MODE4
120 VDU19,0,7,0;:VDU19,1,0;0;
130 PROCass:xs%=4:ys%=4:q$=""
140 DIM T$(2,21),S$(2),name 25
150 PROCldf:F%=FNfont:PROCldp:VDU23,1,
0;0;0;:PROCcrop:PROCparms:PROCprint:m%
=FNmenu
160 IF m%=1 THEN CLEAR:GOTO 130
170 IF m%=2:CHAIN"DESIGNA"
180 END
190 :
1000 DEFFNmenu:VDU26,12:PRINTTAB(3,25)"
MENU"TAB(3)"1. Load format"TAB(3)"2. C
reate format"TAB(3)"3. Quit"":=FNcho
1010 :
1020 DEFPROCwin:VDU28,3,31,39,25,12:PRI
NTq$;:ENDPROC

```

```

1030 :
1040 DEFFNfont:q$=" FONTS":PROCwin:PRIN
TTAB(8)"1. Emphasized"TAB(8)"2. Italics
"TAB(8)"3. NLQ"":=FNcho
1050 :
1060 DEFFNcho:PRINTTAB(6)"CHOICE (1-3):
";:REPEATAS=GET$:UNTILINSTR("123",A$)>0
:PRINTAS:VDU26,12:=VAL(A$)
1070 :
1080 DEFPROCcrop
1090 X%=0:Y%=0:S$(0)="Position box. <SP
ACE> to print.":PROCps:REPEAT
1100 PROCmove(0,1280-D%*4,0,1024-H%*4)
1110 P%=X%+D%*4:Q%=Y%+H%*4
1120 PROCbox(X%,Y%+20,P%-1,Q%,6)
1130 PROCbox(X%,Y%+20,P%-1,Q%,6)
1140 UNTILsp$:VDU7:PROCps:*FX15,1
1150 IF FNyn("Do you want border")=TRUE
:GCOL0,1:PROCbox(X%,Y%+20,P%-1,Q%,5)
1160 ENDPROC
1170 :
1180 DEFPROCparms
1190 D%=D%*2:U%=D%MOD256:V%=D%DIV256
1200 ?&76=P%MOD256:??&77=P%DIV256
1210 ?&70=X%MOD256:??&71=X%DIV256
1220 ?&78=Q%MOD256:??&79=Q%DIV256
1230 ENDPROC
1240 :
1250 DEFFNyn(q$):S$(0)=q$+"? (Y/N)":""PR
OCps:REPEATAS=GET$:UNTILINSTR("YyNn",A$
)>0:PROCps:IF A$="Y" OR A$="y":=TRUE
1260 =FALSE
1270 :
1280 DEFPROCass
1290 dump=&A00:FORpass=0TO2 STEP2
1300 P%=dump:[OPT pass
1310 .linedump LDA&78:STA&72:LDA&79:STA
&73:LDA#0:STA&80:LDA#128:STA&81:.readpix
LDX#&70:LDY#0:LDA#9:JSR&FFF1:LDA&74:BE
Q next:LDA&80:CLC:ADC&81:STA&80:.next CL
C:LSR&81:BCSprinter:LDA&72:SEC:SBC#4:STA
&72:LDA&73:SBC#0:STA&73:JMPreadpix
1320 .printer LDA&75:BEQprint:LDA&80:AN
D#240:STA&80:.print LDA#1:JSR&FFEE:LDA&8
0:JSR&FFEE:LDA#1:JSR&FFEE:LDA&80:JSR&FFEE
E:.nextbyte LDA&70:CLC:ADC#4:STA&70:LDA&
71:ADC#0:STA&71:LDA&70:CMP&76:BCClinedum
p:LDA&71:CMP&77:BCClinedump:RTS
1330 :NEXT:ENDPROC
1340 :
1350 REM Use your printer's own codes
1360 REM here. Omit F%=3 if no NLQ.
1370 DEFPROCfon:VDU1,27,1
1380 IF F%=1 VDU69
1390 IF F%=2 VDU69,1,27,1,52
1400 IF F%=3 VDU120,1,1
1410 ENDPROC
1420 :
1430 DEFPROCfoff:VDU1,27,1
1440 IF F%=1 VDU70

```



```

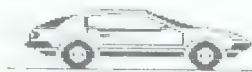
1450 IF F%=2 VDU70,1,27,1,53
1460 IF F%=3 VDU120,1,0
1470 VDU1,27,1,65,1,12,1,13,3:ENDPROC
1480 :
1490 DEFPROCtx(R%):PROCfon
1500 IF (R%=1 OR R%=3) AND B%=0:ENDPROC
1510 IF (R%=2 OR R%=4) AND E%=0:ENDPROC
1520 *FX3,10
1530 IF (R%=1 OR R%=2):PROClin
1540 IF (R%=3 OR R%=4) PRINTSS(R%-2);
1550 *FX3,0
1560 ENDPROC
1570 :
1580 DEFPROClin
1590 A%=-B%*(R%=1)-E%*(R%=2):IF TS(R%,G
%)<>STRING$(A%,"_") THEN PRINTT$(R%,G%);
:ENDPROC
1600 *FX3,0
1610 VDU1,27,1,45,1,1:p$=STRING$(A%," "
):*FX3,10
1620 PRINTp$;:*FX3,0
1630 VDU1,27,1,45,1,0:ENDPROC
1640 :
1650 DEFPROCdraw:VDU1,27,1,76,1,0,1,V%
:CALL dump
1660 ENDPROC
1670 :
1680 DEFPROCclf(Z%):Q%=Q%-Z%*4
1690 ?&78=Q%MOD256:??&79=Q%DIV256
1700 ?&70=X%MOD256:??&71=X%DIV256
1710 VDU1,27,1,65,1,Z%,1,13:ENDPROC
1720 :
1730 DEFPROCprint
1740 VDU2:FOR C%=0TOT%-1
1750 PROCTx(1):??&75=0:PROCdraw:PROCTx(2
):PROCclf(8):IF G%<>T%-1 PROCTx(3):??&75=1
:PROCdraw:PROCTx(4):PROCclf(4)
1760 NEXT:PROCOFF:VDU23,1,0;0;0;0;0:IF
FNyn("Print again?")=TRUE:D%=D%/2:PROCCro
p:PROCParms:GOTO1740
1770 ENDPROC
1780 :
1790 DEFPROCldp
1800 REPEAT:q$="Picture filename (or st
ar cmd)": " :PROCwin:INPUT p$
1810 IF LEFT$(p$,1)=="*" THEN PROCstar E
LSEC%=OPENUPp$:IF C%=0 VDU7:PROCT("No su
ch file!"):PROCdlay(50)
1820 UNTIL C%<>0 AND LEFT$(p$,1)<>"*"
1830 CLOSE#C%:Sname="*LO. "+p$+" 5800":
PROCoscli:ENDPROC
1840 :
1850 DEFPROCps:VDU5:GCOL3,1:MOVE64,96:P
RINTSS(0);:VDU4:ENDPROC

```

```

5050 MOVEK%,L%:PLOT0%,M%,L%:PLOT0%,M%,N
%:PLOT0%,K%,N%:PLOT0%,K%,L%:ENDPROC
5060 :
5070 DEFPROCdlay(S%)
5080 TIME=0:REPEATUNTILTIME>S%:ENDPROC
5090 :
5100 DEFPROCmove(K%,L%,M%,N%)
5110 LOCAL S%,s1%:s%=xs%+(-32*INKEY-1)
5120 s1%=ys%+(-32*INKEY-1)
5130 X%=X%+s%*(INKEY-26-INKEY-122)
5140 Y%=Y%+s1%*(INKEY-42-INKEY-58)
5150 sp%=INKEY-99
5160 IF X%<K% X%=K%
5170 IF X%>L% X%=L%
5180 IF Y%<M% Y%=M%
5190 IF Y%>N% Y%=N%
5200 ENDPROC
5210 :
5220 DEFPROCoscli:LOCAL X%,Y%
5230 X%=name MOD256:Y%=name DIV256
5240 CALL &FFF7
5250 ENDPROC
5260 :
5270 DEFPROCldf
5280 REPEAT:q$="Load format filename (o
r star cmd)": " :PROCwin:INPUT p$
5290 IF LEFT$(p$,1)=="*" THEN PROCstar E
LSE C%=OPENUPp$:IF C%=0 VDU7:PROCT("No s
uch file!"):PROCdlay(50):CLOSE#C%
5300 UNTIL C%<>0 AND LEFT$(p$,1)<>"*"
5310 INPUT#C%,D%,H%,T%,B%,E%
5320 FORi%=1TO2:INPUT#C%,SS(i%):NEXT
5330 FORi%=0TOT%-1:INPUT#C%,TS(1,i%),TS
(2,i%):NEXT:CLOSE#C%
5340 PROCT("Format loaded."):PROCdlay(5
0):ENDPROC
5350 :
5360 DEF PROCstar
5370 VDU26:CLS:Sname=p$:PROCoscli
5380 PRINT""Press spacebar to continue"
";
5390 IF GET CLS:ENDPROC
5400 :
5410 ON ERROR OFF
5420 MODE7
5430 REPORT:PRINT" at line ";ERL
5440 *FX3,0
5450 END

```



```

5000 REM SHARED LINES Ver 0.2
5010 :
5020 DEFPROCt(q%):PROCwin:VDU26:ENDPROC
5030 :
5040 DEFPROCbox(K%,L%,M%,N%,O%):*FX19

```

Pursuing the Trivial

Title : Trivial Pursuit
Supplier : Domark
Price : £14.95 (includes two tapes)
Rating : ***
Reviewer : Sarah Williams

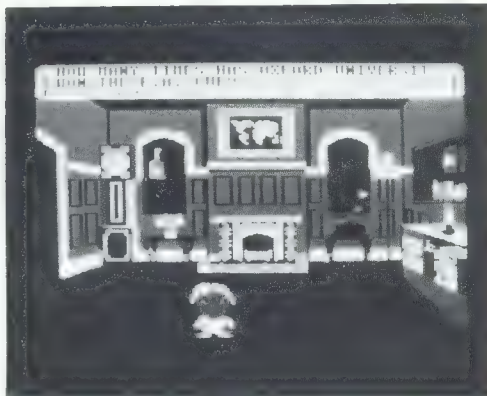
After the huge success of the board game "Trivial Pursuit", it was perhaps inevitable that a computer version would eventually appear. Domark is the company concerned and they are already claiming enormous interest in the computer version.

First impressions of the game were coloured by the fact that the instructions were neither informative nor clear. Indeed, it took quite some time to get the game started. Once the game is underway you are introduced to the character of T.P. who guides you through the quiz and acts as a referee.

Unlike many computer games, where you play against the computer, you have the option of playing against as many as five opponents. Of course, if you wish, you can just practice in 'single-player' mode. As in the original board game, those playing take it in turns to answer questions on a variety of subjects. In the computer version, unlike the board game, these questions include both visual picture questions and audible music questions.

When you are ready to begin playing, T.P. randomly throws a dart at a number from one to six. This is a substitute for the old fashioned die. Your choice of squares to move to is then highlighted on the screen. When you have decided on your chosen subject the screen changes and you find yourself in T.P.'s study. Depending upon the question category selected, the study is decorated appropriately. When a graphical question arises the lights dim and a screen rolls down from the ceiling ready for the projected image. Musical questions are heralded by T.P. walking over to the tape recorder.

Once you have worked out your response you press the space bar and the correct answer is displayed on the screen. It is



then up to you and your opponents to decide whether you were correct or not. One feature you will either love or hate is the way T.P. paces impatiently up and down, audibly tapping his feet, as he waits for your answer.

The graphics of the game, to my mind, could have been better in places. The board itself on a colour monitor was clear enough, but on a monochrome monitor it was difficult to determine which 'colour' belonged to which subject category.

The display gives an adequate representation of the game most of the time, but on any visual questions the pictures are quite small with little detail, and in several cases unrecognisable in my view.

Overall, if I was to compare it to the original board game, I would have to favour the latter. The computer version seems to introduce too many unwanted complications that detract from the spontaneity and enjoyment of the original. The continual change of scene from the game board to the study and back to the board also becomes a little tedious after a while. The only advantage of the computer version is the opportunity to include musical questions.

Despite many irritations in the playing of the computer version of Trivial Pursuit, the name alone is probably sufficient to ensure that Domark will have

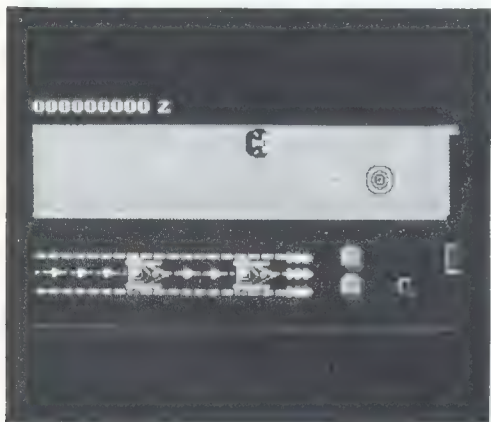
**• GAMES REVIEWS • GAMES REVIEWS • GAMES REVIEWS • GAMES REVIEW!
GAMES REVIEWS • GAMES REVIEWS • GAMES REVIEWS • GAMES REVIEWS**

a runaway success this coming Christmas. But in my view, the original board game will provide far more enjoyment for the majority of players.

Psycastria

Title : Psycastria
Supplier : Audiogenic
Price : £7.95
Rating : *****
Reviewer : Daniel Gaster

"Superfast smooth scrolling shoot-'em-up action", states the blurb on the packet. Psycastria is one the fastest, most difficult and addictive games to be released for the Beeb.



The idea of the game is to fly above the surface of four different 'installations' (on land, sea, the moon and in space) destroying energy pods. The screen windows a section of the current level, which scrolls smoothly from side to side. You control a spaceship that can travel left, right, up and down. Some buildings rise above the surface and so must be avoided; other obstacles such as fighter planes can be destroyed for extra points. The best part of Psycastria is the graphics which, despite being shown on a rather small screen, are colourful and detailed.

The presentation, sound effects and playability are all excellent. My main complaint is that Audiogenic say that the game (tape version anyway) will not load with Replay fitted. Overall this is a top-class game. Very addictive and challenging and worth buying even if you're not a shoot-'em-up fan.

Galaforce

Title : Galaforce
Supplier : Superior Software
Price : £9.95 Tape
: £11.95 Disc
Rating : *****
Reviewer : Daniel Gaster

Galaforce is a fast-moving, colourful, all-action game in the Zalaga mould. You can move your spaceship in eight directions and the inevitable fire button can be held down for maximum blast power. The idea of the game may seem hauntingly familiar but this is no boring, run-of-the-mill game. There are seemingly endless streams of attacking aliens.



The game is divided into Zones, each Zone consisting of several waves of nasties, swooping down in complicated formations. Some require as many as ten hits to kill them. At later stages the aliens drop a deadly assortment of bombs.

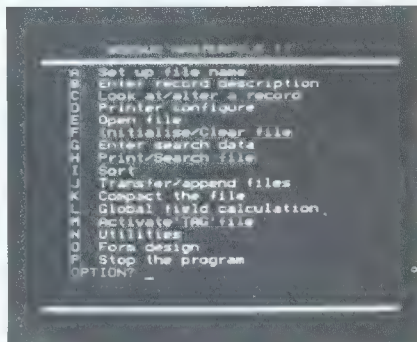
→ 47

MASTERFILE II — General purpose database

Combines with Wordwise/View to produce Standard Letters etc.

MASTERFILE is a general purpose file management package allowing large amounts of information to be stored and processed. It is extremely powerful yet flexible and easy to use. Once set up, the information may be retrieved, sorted on any field, displayed, updated, printed etc. as required.

Separate versions of the program are available for cassette and disc users, the disc version using random access files and offering many extra features.



MASTERFILE II FACILITIES

- Extremely fast "tag" sorting on virtually any combination of fields. Any number of tag files may be associated with your data file.
- Flexible print layouts allowing additional text to be combined with the output to produce standard letters etc.
- Output from MASTERFILE II may be spooled directly to a file for later use with WORDWISE or VIEW.
- Flexible data entry.
- Fields may be defined as numeric, string, decimal or date and up to 255 characters in length.
- Spreadsheet type facilities allowing global changes to be made on string, numeric and decimal fields.
- Sub-files may be created from selected fields and records, from the master file, according to most criteria.
- Extra utilities to delete a field throughout a file, copy file descriptors and change dates.

ORDERING INFORMATION

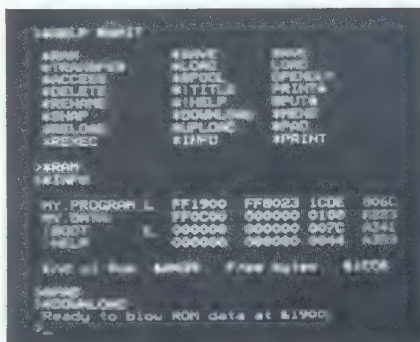
	Price	Mem Price
Masterfile II	£22.00	£16.50
ADFS Masterfile II	£22.00	£16.50
Upgrade M.file II to ADFS M.file II	£7.00	£5.00
Upgrade M.file I (Disc) to M.file II	£12.50	£9.35
Upgrade M.file I (Disc) to ADFS M.file II	£12.50	£9.35

Please specify 40 or 80 tracks and standard or ADFS format.

ROMIT — Put your own basic programs into Rom

- Create Your Own Silicon Disc
- Uses Sideways or Normal Ram
- A Complete Filing System with up to 16K capacity (per bank of Ram)

ROMIT adds a new feature to your BBC Micro — A Ram Filing System. This provides over 20 new commands which enable you to:



1 Create a silicon disc with its own filing system commands. Type **★RAM**, then you can catalogue the 'disc', save and load files to it, storing as many files as memory will allow. **★DISC** (or **★TAPE**) takes you back to your normal filing system.

2 Put your own Basic or assembler programs (or any files into Ram and then make them into Eproms. Whether it's your favourite game, your function key definitions or any program that you use frequently, you can now put it into Eprom so that it's always instantly available. Multiple copies can also be produced to enable you to pass on your programs to other people.

Getting your own Basic program into Rom couldn't be simpler. Just load your program into memory and type:

★RAM ★FILE 'prog' ★BLOCK

and either connect it to a blower, or send the disc (or tape) to us, together with your **Romit** registration number, and we will Rom it for you. (There is a small charge for this service).

3 Use your sideways Ram as a printer buffer so that you can continue to use your computer for other work at the same time as printing long documents.

OVER 20 NEW ★COMMANDS ARE PROVIDED BY ROMIT

An Eprom Blowing Service is offered with this pack.

To get the most from Romit you will need sideways Ram fitted to your computer, eg just a single Ram chip plugged into your ATPL (or other) Rom board.

**MEMBERS
PRICE
£25.50**

ADVERTISING IN BEEBUG

For advertising details, please contact:

Lynn Lloyd Yolanda Turuelo
on or on
(0923) 677222 (0727) 40303

or Write to:
Dolphin Place
Holywell Hill
St. Albans
Herts AL1 1EX

MIRRORSOFT

BY MAIL ORDER

The entire Mirrorsoft range of quality software is available by mail order direct from our warehouse with no extra charge for postage and packing (UK only). All programs run on BBC B B + and Electron. Please enquire about Master-compatible versions.

Prices given are for tape versions. Please add £3 for 5.25" disks. Please state whether 40 or 80 track required.

We accept cheques, postal orders, Access, VISA, and American Express. Please allow 28 days for delivery. Send for our free colour catalogue, too!

EARLY LEARNING/CHILDREN'S ENTERTAINMENT

Caesar's Travels	£12.95 (disk only)	Ages 3-9
Count with Oliver	£7.95	Ages 4-7
Crack It! Towers	£9.95	Ages 7+
First Steps with the Mr Men	£8.95	Ages 3-6
Giddy Game Show	£9.95	Ages 3-6
Here & There with the Mr Men	£7.95	Ages 4-7
Look Sharp!	£7.95	Ages 5-11
Mr Men Magic Storymaker	£9.95	Ages 4-7
Quick Thinking!	£6.95	Ages 5-11
Quick Thinking Plus!	£6.95	Ages 5-11
Word Games with the Mr Men	£9.95	Ages 5-8

CREATIVE COMPUTING

Fleet Street Editor	£39.95 (disk only)
Admin Xtra	£14.95 (disk only)
Fonts 'n' Graphics	£14.95 (disk only)
Walt Disney Graphics	£14.95 (disk only)

GAMES

Caesar the Cat	£6.95
Hi Bouncer	£6.95
Strike Force Harrier	£9.95

HOME DISCOVERY

BBC Mastermind	£9.95
BBC Mastermind Quizmaster	£5.95
Joffe Plan	£9.95
Know Your Own Psi Q	£9.95
Know Your Own Personality	£9.95
Star Seeker	£9.95

MIRRORSOFT

Dept BB,
FREEPOST BS4382,
Paulton, Bristol BS18 5BR (No stamp needed)

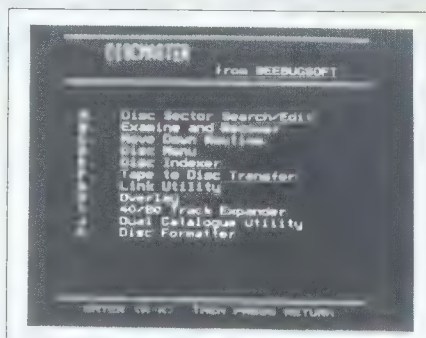
DISCMASTER

— Utilities for the disc user

Eleven powerful utilities to support the disc user, with routines to help with just about everything you'll want to do while using your disc drive.

All programs are called from a main menu and are explained in detail in the comprehensive manual.

Members Price £15.75



- **Disc editor/String search**

A string search routine combined with a full screen editor, simply overtype on the screen (in hex or ascii) to update the disc.

- **Recover**

Recovers programs lost by accidental deletion or disc corruption.

- **Movedown**

The command ★RUN will load, move down and run your long programs automatically, if you add this routine to them.

- **Disc menu**

A colourful automatic menu to add to each of your discs, which will move down and RUN Basic programs, and ★RUN machine code programs.

- **Indexer**

Creates an information index file for all of your discs, which may then be displayed, updated or printed to help you keep track of your programs.

- **Copier**

Transfer a whole cassette of programs onto disc automatically. (Will not copy protected cassettes.)

- **Link**

A utility to automatically call and insert useful routines from your own library into your development programs.

- **Overlay**

Enables procedures or functions to be loaded from disc and overlaid into your program as it runs, allowing you to run programs much larger than available memory.

- **40 to 80**

Insert your 40 track disc. 3 minutes later remove the disc which is now in 80 track format and still has all of your programs on it.

- **Dual catalogue**

Enables 61 files to be saved per disc surface instead of just 31.

- **Format**

A speedy program to format or verify 40 and 80 track discs.

"A godsend to disc users"

EDUCATIONAL COMPUTING

"Some very interesting facilities"

PERSONAL COMPUTER NEWS

COMING SOON FROM BEEBUGSOFT

BEEBUGSOFT COMMUNICATIONS SOFTWARE

A completely new Rom offering over 50 commands to open up the world of communications with the Beebug Magic Modem. This package contains all the usual features such as Dowland, Text Terminal, Viewdata Terminal, but offers many new exciting features.

- All features are available with ★ commands such as ★CALL, ★ANSWER, ★DOWNLOAD. Basic programmers can now write their own powerful communications software very easily without resorting to machine code.
- Telephone directory in rom allows you to enter the name, number, modem configuration etc. for easy recall at a later date. No need to remember telephone numbers any more, just enter the name instead of the number e.g. ★CALL 'prestel.'
- Other special features include an auto-answer facility, a very powerful teletext frame editor with help facility, a real time clock, and much much more!

Bulletin Board Software, Enhanced Telephone Directory and Utility programs coming soon on disc.

NEW BEEBUGSOFT UTILITY ROM

A new and exciting utility Rom specially developed for the Master 128 (although most features are compatible with the BBC B, B+ and Master Compact). Features include:

- An automatic disc menu giving complete control over disc files, and especially useful to ADFS users for easy movement through ADFS directories. Files can be loaded, run, or marked for deletion, copying, renaming etc. This is an exceptionally useful utility containing many powerful options, and once installed becomes an indispensable aid for serious disc users.
- An excellent visual display of computer status allowing settings to be displayed, edited, loaded and saved. Gives complete control over sideways Roms, screen status, free memory etc. and combined with the Automatic Disc Menu provides a very powerful piece of software.
- A comprehensive diary package for keeping track of appointments and important dates. Includes a monthly calendar display, diary, clock, automatic alarm and many other features. Plus a complete range of powerful disc, sideways ram and general purpose commands — further details next month.

**FURTHER DETAILS
NEXT MONTH**

Classified Ads

Problems? try R-SOFT utilities!

1. HOW-TO: An essential collection of software and instructions for frustrated new disc owners who want to move their programs to disc.

3. ROMPULL + TAPEDUMP.

4. AUTOMATIC DISC MENU: Includes sideways-RAM version, can boot from ROM, works with 2nd 6502.

6. SWROM*: Puts your Basic / machine code programs in ROM format.

7. RFS-GENERATOR: Generates ROMs for the *ROM filing system. This does not use the DFS workspace and is an ideal tool to run nasty programs from disc.

All above packages £5.00 each.

13. D-MASTER-V2: Superb disc backup program; needs 8271 and sideways RAM; £8.00; on 16K EPROM £12.00. Upgrade for D-MASTER owners: £5.00/£9.00.

16. OPUS SD/DD/Challenger: Disc menu, disc editor, disc indexer, etc. Please enquire for further details.

17. D-EDITOR: based on D-MASTER-V2: will edit any disc D-MASTER-V2 can backup! £8.00/£12.00.

Please send SAE for full list.

R-SOFT, 22 Marriotts Close, Felmersham, Bedford MK43 7HD. Tel: (0234) 781730.

Disc drive, half-height, D/S D/D 40/80 track switchable. New and guaranteed. Uncased. £75. Cased with PSU £95. Tel: (06845) 68676 evenings.

CONVERT your 80T drive to 40/80 switchable. MkII unit (cased) simply plugs in - £19. MkI kit as above but requires soldering iron, fits inside drive - £11. Sideways RAM system 16K with software to load/save ROMs - £25. All prices include VAT and p&p. Computer repairs, competitive rates. Please telephone for further information. Dept B2, MAGUS Electronics, 33 Pool Lane, Winterley, Sandbach, Cheshire CW11 0RZ. Tel: (0270) 582748.

MARKS AND STATISTICS For teachers and lecturers. Simple to use spreadsheet format. 330 students per file. Up to 300 subjects. Sorts, analyses, normalises, calculates means, applies weighting factors. Prints histograms and lists. Full documentation supplied. Disc (40 or 80 track) £17. In-Form, 73 Woodfield Park, Colinton, Edinburgh EH13 0RA.

Members' Corner

Mr. P. S. Flowers of 12 Sevenacres, Somerton, Somerset has a problem in using his UDM double density disc filing system. He is trying to perform double stepping using an OSWORD call with A set to &72 (not &7F) but with no response. Anyone who may be able to help Mr. Flowers can contact him at the above address or phone him on 0458-73144.

Events

Electron and
BBC Micro User

(Discount off entrance — see voucher in this issue)

Higher Technology
and Equipment in
Education

Royal Horticultural
Halls, Westminster

Barbican Centre
London

7 - 9 Nov

21-24 Jan

BEEBUG

at the



**Royal Horticultural Hall, Westminster, London
7th - 9th November 1986**

**Visit BEEBUG and BEEBUGSOFT at the Micro User Show
- Stand 87**

- ★ 50p off standard show entrance price using voucher below.
- ★ Special half-price offers on many back issues of BEEBUG magazine.
- ★ Special offer on BEEBUG dynamic blank discs.
- ★ Many other special show offers.
- ★ All magazine cassettes and discs, including Magazine Games and Magazine Utilities.
- ★ Full range of BEEBUGSOFT products.
- ★ Magazine and technical staff to answer all your queries.



U.K. Bulletin Boards

All the above listed boards were active as of late September 1986. Circumstances may have changed since then and BEEBUG would appreciate members letting us know of any changes they have discovered such as boards that have ceased operating or changed times and baud rates. Also, if you know of a board that is not listed here, please let us know for the benefit of other members.

Some new boards, particularly NBBS and OBBS systems, enable Beeb owners with Commstar or the Demon Zromm to have colour. With Commstar use Mode 7 and switch off the filter mask at the command screen before entering 'Chat' mode. Zromm users should use Mode 7 and *Chat instead of *Terminal. Then when logged on answer 'Yes' to the question 'Are you using software on a BBC that allows colour'.

Aberdeen ITEC 24 Hours	0224 641585 (Aberdeen) 1200/75 Viewdata	Brixton ITEC 24 Hours	01 735 6153 (London) 1200/75 Viewdata
Acorn 24 Hours	0223 243642 (Cambridge) 1200/75 Viewdata	C-View 24 Hours	0702 546373 (Essex) 1200/75 Viewdata
Amrad 24 Hours	021 744 1558 (Solihull) 300/300	CBABBS 24 Hours (Ex.Thurs.)	021 430 3761 (Birmingham) 300/300
BABBS 21.00-00.00	0225 23276 (Bath) 300/300 *	CBBS London West 24 Hours	0895 420164 (London) 300/300
BABBS I 24 Hours	0394 276306 (Felixstowe) 300/300	CBBS Surrey 24 Hours	04862 25174 (Surrey) 300/300
BABBS II 24 Hours	0268 778956 (Basildon) 300/300	CBBS South West 24 Hours	0392 53116 (Exeter) 1200/75 & 300/300
Basildon ITEC 24 Hours	0268 22177 (Basildon) 1200/75 Viewdata	CFC#9 24 Hours	0395 272611 (Exmouth) 300/300
Basildon ITEC 24 Hours	0268 25122 (Basildon) 300/300	CNOL 24 Hours	0524 60399 (Lancaster) 300/300
Belton BB 24 Hours	0493 781334 (Great Yarmouth) 1200/75 & 300/300	COCO BBS 24 Hours	0912 651944 (Tyneside) 300/300
Bloxham NOBB 22.00-01.00	0295 720812 (Banbury) 300/300	Communitel 24 Hours	01 968 7402 (London) 1200/75 Viewdata
Bradford OBBS 18.00-00.00	0274 496783 (Bradford) 300/300	Communitree 21.00-08.00	0874 711147 (Wales) 300/300

Personal Ads

BEEBUG members may advertise unwanted computer hardware and software through personal ads (including 'wants') in BEEBUG. These are completely free of charge but please keep your ad as short as possible. Although we will try to include all ads received, we reserve the right to edit or reject any ad if necessary. Any ads that cannot be accommodated in one issue will be held over to the next, so please advise us if you do not wish us to do this. We will accept adverts for software, but prospective purchasers should ensure that they always receive original copies including documentation to avoid any abuse of this facility.

We also accept members' business ads at the rate of 30p per word (inclusive of VAT) and these will be featured separately. Please send all ads (personal and business) to MEMBERS' ADS, BEEBUG, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX. The normal copy date for receipt of all ads will be the 15th of each month.

BBC B, O.S.1.2, Basic II - £225 ono.
6502 second processor - £100 ono. HCR
RAM/ROM box (2 boards 32k RAM) - £100
ono. ROMs - Disc Doctor, Floppywise
Plus, Enigma. Robin, Tel:(0543) 254805
after 6pm.

32K shadow RAM board, ROM and manual
(Watford) £40. Tel:(02403) 4206.

BBC B. Acorn DFS, 100K drive, Watford
32K RAM card, View 2.1, Toolkit, BEEBUGS
complete, Advanced user and disc guides,
books mags and software. £500 the lot.
Aldershot (0252) 316712.

BBC B, Acorn DFS 1.2, 2x400K drives,
Watford 32K shadow RAM, ATPL ROM board +
16K sideways RAM. AMX Super Art + mouse
+ Inter-word, Inter-sheet, ISO Pascal,
BCPL, Comal, GKR and many more ROMs all
with original manuals. Some games.
Prepared to split hardware and software.
£655. Full set of Acorn Users £20 and
BEEBUG £15. Tel: Hatfield (07072) 64378
evenings.

For sale: BBC slimline 40 track SS 100K
disc drive, UDM double density disc
controller, Beebugsoft Masterfile II (40
track) £50. Phone (0582) 881614.

Quendata DWP 1120 users. *commands,
details. 22 Bramley Drive, Offord
D'Arcy, Huntingdon PE18 9SF. Tel:(0480)
812017 evenings.

For sale, Electron Plus 3 disc drive and
interface, View cartridge, disc database
and 10 blank 3.5 inch discs. £150. Phone
Phil Jones (0533) 56037 ext 277 during
business hours.

Laserbug magazines wanted to complete
project. Details to 101 St.Nicolas Park
Drive, Nuneaton, Warwickshire CV11 6DZ.

BBC B with Acorn DFS and sideways ZIF
socket plus single 80 track double sided
disc drive. £250. Phone Spalding (0775)
3823.

For Sale. Wordwise Plus ROM,
introduction and reference manuals
complete with cassette typing tutor.
£35. Shepherd, Milton Keynes (0908)
313379.

BBC B, O.S.1.2, Opus DFS, 100K/175K
single side slim disc drive, manuals.
£285. R.Smith, 60 Mendip Ave., Weston-
Super-Mare BS22 0HT. Tel:(0934) 513934.

Pan Course Tutors Study and Revision 'O'
level packs complete with books and
guide: Physics (three discs) £10.00 -
Biology (two discs) £9.50. Also 'BEL
BASE 2' disc based database (latest
version) £13.50 and 'BEL CAD' with
character and icon designer £10.00. All
complete with manuals in good condition.
W.J.Lambert, Malt Cottage, Main Street,
Wymondham, Melton Mowbray, Leicester-
shire LE14 2AG. Tel:(057) 284664.

Viglen PC console £38; StarDatabase ROM
and utilities £40; ATPL sideways ROM
board £30; Beebugsoft Paintbox disc £6;
Vine Replay ROM 8271 £27; Snapper disc
£5; Carousel disc £5; BBC dust cover £2;
Disc drive dust cover £2; All mint
condition. Tel:(031) 6641809.

Chartbusters, Combat Lynx, Wizardore, 3D
Grand Prix, Jetpac - £4 each, Overdrive,
Dune Rider, Croaker, Moonraider, Gold
Digger - £2 each. Marlow (06284) 6745.

Acornsoft/Mi business software series.
Invoicing and mailing system packages
(40 track discs) £10 each. Tel:(0303)
42540 evenings.

Compost Heap 24 Hours	0622 46036 (Maidstone) 300/300 1200/75	Hackney BB 24 Hours	01 985 3322 (London) 1200/75 Viewdata
Cyberzone 24 Hours	01 638 2034 (London) 300/300	Hamnet 17.30-08.45 Mon-Sun 12.30-13.45 Mon-Fri	0482 465150 (Hull) 300/300
Cymrutel 24 Hours	0492 49194 (Wales) 1200/75 Viewdata	Haunting Thunder 21.00-08.00	0752 364059 (Plymouth) 300/300
Distel 24 Hours	01 679 1888 (London) 300/300	Health Data 24 Hours	01 986 4360 (London) 1200/75 Viewdata
Distel 24 Hours	01 679 6183 (London) 1200/75	Inferno 22.00-23.30	0206 41401 (Essex) 300/300
Electronic Tourist Board 24 Hours	0874 730172 (Powys) 300/300 1200/75	ITCU Exchange & Mart 24 Hours	01 960 4742 (London) 1200/75 Viewdata
Fastnet Fido 22.00-08.00	051 260 5607 (Liverpool) 300/300 (Bell)	Keyboard 20.00-00.00	0908 668398 (Milton Keynes) 300/300
FBBS Jersey 18.00-06.00	0534 55855 (Jersey) 300/300	Livingstone BBS 24 Hours	0506 38526 300/300
FBBS Swansea 24 Hours	0792 203953 (Swansea) 300/300	London BB 24 Hours	01 455 6607 (London) 1200/75 & 300/300
Fido - UK 15.00-06.00	0635 46480 (Berkshire) 300/300	London W Tech Centre 24 Hours	0895 52685 300/300
Forum 80 London Evenings (Ring & ask)	01 902 2546 (London) 300/300	London Underground 24 Hours	01 863 0198 (London) 1200/75 & 300/300
Forum 80 Hull 19.30-23.30	0482 859169 (Hull) 300/300	Mailbox 80 24 Hours	051 428 8924 (Liverpool) 300/300
Forum 80 Spa 22.30-00.30	0926 39871 (Leamington Spa) 300/300	Maptel 24 Hours	0702 552941 (Essex) 300/300
Gnome at Home 24 Hours	01 888 8894 (London) 1200/75 Viewdata	Marctel 10.00-22.00	01 346 7150 (London) 300/300
Gosport Apricot 18.00-00.00	0705 524805 (Gosport) 300/300	MBBS Leconfield 24 Hours	0401 50745 (Beverley) 300/300

TO BE CONTINUED

ease. On the BBC micro the 32 characters with codes from 224 to 255 are reserved for redefinition. On the Master and Compact all characters in the range 32 to 255 apart from 127 (delete) are held in RAM and are redefinable directly.

As an example of character redefinition, try the following in any of the graphics modes (0-6):

```
VDU23,255,60,126,219,255,126,60,36,66
PRINT CHR$(255)
```

This will print a space invader character, and this technique is used very heavily in games programs of all sorts to put character images anywhere on the screen. The TAB command can then be used to position your invader or whatever, and if you wish to simulate movement, then a simple FOR-NEXT loop, altering the TAB position, will achieve this. But do not forget to erase the previous image when printing up the next, or you will end up with a continuous trail of invaders.

In this example, we redefined character 255 as the invader character, but you may also redefine currently used characters. For example, the following will redefine the normal prompt character used in modes 0-6 as an invader:

```
VDU23,62,60,126,219,255,126,60,36,66
```









And the following will redefine the space character (code 32) as a solid box:

```
VDU23,32,255,255,255,255,255,255,255,255
```

This allows you to see if you have any extra spaces in program lines, or in a piece of text (providing it is displayed in some mode other than 7 or 135). Try issuing this call, and then listing a program, or using Wordwise in preview mode, or View (but not in mode 7 or 135).

We have seen that VDU23 can be used to create new character designs. But how, precisely do these strings of digits define a new character? The answer is simple for all those off-world beings who compute in binary. The first number after the VDU23 is the number of the character to be redefined - character 62 for the prompt or 32 for a space. The other eight numbers represent the state of each of the 64 pixels on an 8 x 8 character grid.

As you can see from the figure, the first row of bits is represented by 00111100. Each "0" means that the corresponding point on the grid is off,

Invader Character	Binary Code	Decimal Code
	00111100	60
	01111110	126
	11011011	219
	11111111	255
	01111110	126
	00111100	60
	00100100	36
	01000010	66

Character Definition:

```
VDU23,60,126,219,255,126,60,36,66
```

and each "1" that it is on. The string of digits 00111100 is taken by the computer to be a binary number which represents a row of the character definition. Unfortunately we cannot talk to the computer in

binary, and we have to do the donkey work of translating this binary number into decimal (or hexadecimal) for the character definition. Translated into decimal, 00111100 is 60, and this is the second parameter in the VDU23 character definition for the space invader. A further 7 parameters inform the computer of the desired pattern for the remaining 7 rows of the character grid.

You may well have a character definer program which does all this work for you, but you may also find the accompanying listing (Program 1) of use. It converts binary numbers to decimal, and thus considerably speeds up the character definition process. It will convert any binary number of up to 31 digits in length, and prints out both the decimal and hexadecimal equivalent. The program incorporates an input trap, and will ignore entries containing non-binary characters. If you do not need this, you can considerably shorten the program: just ignore lines 20, 50 and 120 - 180.

```
10 REM PROGRAM1: Binary to Decimal v5
20 REPEAT
30 REPEAT
40 INPUT "Binary number "AS
50 UNTIL FNTEST
60 N=0
70 FOR A=1 TO LEN(AS)
80 N=N-2^(LEN(AS)-A)*(MID$(AS,A,1)="1")
90 NEXT
100 PRINTN;" dec";SPC5;"&";"N;" hex"
110 UNTIL FALSE
120 :
130 DEFFNTEST
140 TEST=TRUE
150 FOR A=1 TO LEN(AS)
160 IF MID$(AS,A,1)<>"1" AND MID$(AS,A,1)<>"0" THEN TEST=FALSE
170 NEXT
180 =TEST
```


READING CHARACTER DEFINITIONS

The computer's character definitions, or dot patterns as we called them earlier, are stored in exactly the same format as used in the VDU23 character definition call: that is to say, as a sequence of 8 numbers representing the 8 x 8 character grid. A special call is provided for the user to read the resident character definitions, and this can be useful if you wish to manipulate the characters, or change their scale in any way.

The process of reading the definitions involves a so-called OSWORD call to the operating system. But we may ignore the ins and outs of this by parcelling up the call in a short procedure. This is listed in Program 2. The procedure itself appears from line 1000 onwards. The earlier part of the program just gives an example of its use. Lines 120 and 130 set up a memory

```
100 REM PROGRAM2: Character Read PROC
110 REM with Example v7
120 D=&A00
130 DIM B(8)
140 MODE4
150 PRINTTAB(8,3)"CHARACTER INVERTER"

160 REPEAT
170 PRINT"Press any key ";
180 C=GET
190 PROCCHREAD(C)
200 VDU23,253,B(7),B(6),B(5),B(4),B(3),
B(2),B(1),B(0)
210 PRINTCHR$(C);SPC2;CHR$(253)
500 UNTIL FALSE
510 :
1000 DEF PROCCHREAD(I)
1010 ?D=I
1020 X%=0:Y%=&A:A%=&A
1030 CALL &FFF1
1040 FOR A=0 TO 7
1050 B(A)=D?(A+1)
1060 NEXT
1070 ENDPROC
```

area for the OSWORD call, and dimension an array B(n), which will hold the result of the read operation. These two lines should be included in any program calling the character-read procedure. Line 180 uses the GET function to return the ASCII value of any key pressed. This is then passed as a parameter (C) to the procedure PROCCHREAD in the following line. This reads the definition of the character of ASCII code C, and places the result in the array B(n) in such a way that B(0) holds

the number representing the top row, B(1) the next, and so on down to B(7).

Line 200 then defines character 253 using the 8 values read by the procedure, but as you can see, it does this backwards, using the bottom line (B(7)) first. The result of this is that character 253 holds an inverted version of the original. The two versions are then printed out in line 210.

CHARACTER REVERSAL

Once we have read a character definition using PROCCHREAD there are many things that we can do with it. Program listing 3 contains a few lines which may be added to listing 2. They cause two further images of the keyed character to be produced. They are reversed out versions of the two images produced by listing 2. The colour reversal is achieved thanks to a simple property of binary numbers. Each of the 8 numbers of the character definition is subtracted from 255 (11111111 binary). This produces what is called the binary complement of the number. In other words, for each binary one, a binary zero appears, and vice versa. An example should make this clearer. We will take the binary complement of the number 60, used in the first row in the space invader definition:

```
11111111 - 00111100 = 11000011
255      -   60      =   195
```

A reversed out space invader would have 195 as the value of its first row. This reversing process is carried out in the FOR-NEXT loop in lines 230 - 250, and the two new characters defined in lines 260 and 270. Line 280 then prints out the two new images.

```
220 REM REVERSE OUT CHARACTER v1
230 FOR A=0 TO 7
240 B(A)=255-B(A):REM REVERSE CHR
250 NEXT
260 VDU23,254,B(0),B(1),B(2),B(3),B(4),
B(5),B(6),B(7)
270 VDU23,255,B(7),B(6),B(5),B(4),B(3),
B(2),B(1),B(0)
280 PRINT"TAB(15)CHR$(254);SPC2;CHR$(255)
```

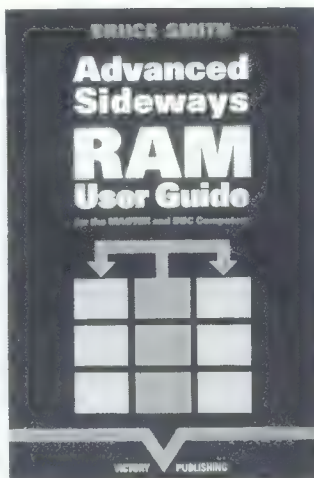
Next month we will continue this discussion, and show how to create double height, double width and rotated characters, as well as giant text.

SIDEWAYS RAM EXPOSED

Will this latest book by Bruce Smith live up to its title and delve deep into the inner workings of sideways RAM? Our own expert in this field, Bernard Hill, gives his considered view.

Advanced Sideways RAM User Guide by Bruce Smith, published by Victory Press at £9.95.

With a book on sideways ROMs and RAMs already on the market by this author ("The BBC Micro ROM Book" - see review in BEEBUG Vol. 4 No. 3) the immediate question is "so what's different about this one?". A glance at the subtitle ("for the Master and BBC Computers") gives a clue. In fact this book is badly subtitled - there is precious little for the BBC micro user: throughout the text the examples are aimed directly at the Master without any warning - Welcome discs, Private RAM, *SRLOAD etc., and the programs are wrongly said to be Basic II compatible (the vast majority use 65C02 assembler instructions). And while we're on the negative side let me purge myself of the book's most annoying trait: it's been word processed on a daisy wheel printer without a distinctive figure "1", so every single occurrence of this number shows a letter "1" overprinted with an up-arrow ^: totally distracting when reading. In fact the whole book has the taste of inadequate preparation and proof-reading. Printing errors are common, (page 133 even contains a typed-in proof-reader's comment!), the index is quite inadequate (no entry for "memory", "load", "link"...), and occasionally there are minor factual inaccuracies. To call it a "User Guide" is also rather misleading as it is in no way an "official" book, and "Advanced Sideways RAM Tutorial" might have shown better its strengths.



The first chapter is the usual difficult compromise in addressing both beginner and expert. The real core of the book is contained in chapters 2 to 15, and 5 appendices. Chapter 2 is a clear explanation of the sideways header, and is an expanded 'Master' version of the corresponding chapter of the earlier book. With chapters 3 to 5 we have a much fuller explanation than previously of the notion of service calls, and in particular *HELP and * command interpreters, (with clear lists of new service calls particular to the Master). The order of

the previous volume is much improved here, and the inclusion of a chapter on defining your own OSBYTE and OSWORD calls (Chapter 9) is an enlightening divergence from the standard approach. A short chapter on extended vectors (5 pages!) is followed by a round-up of other service calls, and Chapter 9 is a very nice addition on extending *CONFIGURE and *STATUS options via a sideways ROM. A brief examination of auto-booting is followed by a much-too-short chapter on workspace: difficult enough on the BBC, but with all the hidden workspace on the Master 7 pages is not really enough! The very clear chapter on the Rom Filing System contains essentially the same RFS formatter as the earlier volume and chapters on Languages and Error Handling are very clear.

Bruce has improved over the earlier book in that he has discarded the soon out-of-date software and hardware reviews and expanded the listings to 25 very clear programs. Whether this book is for you very much depends on what else you have: the proficient SWR programmer with all the Master User Guides will find little of use, but for most lesser mortals the collection of programs together with list of new OSBYTE/OSWORD service calls in one volume might well save the £50 on User Guides alone. Once you get over the unpolished style of the book it makes fascinating and challenging reading.

This book is available from BEEBUG - see mail order price list.



MASTER
SERIES

Executive
Power

Automated path selection, instant backups and an infinitely extendable Help option are just some of the things that can be implemented on the Master and Compact using EXEC files. David Graham shows how it's done.

When an EXEC file is called, its contents are treated as if they were entered from the keyboard. For those familiar with the IBM PC, they are similar in many respects to MS-DOS batch files. Thus if you create an EXEC file which contains the two lines:

```
*KEYØ LOAD
```

```
*WORD
```

this would set up function key Ø with the word LOAD, and then call up the View wordprocessor. But we can create much more powerful EXEC files than this, as I hope to demonstrate in a moment.

EXEC files are made the more useful because of two special features of the Master and Compact. Firstly, you can EXEC into the computer any suitable file by simply preceding its filename with an asterisk. Thus *H will EXEC in a file named "H". This feature of the 177Ø DFS and ADFS makes it very easy to call EXEC files. Model B users must precede the filename with *EXEC (or *E.). Secondly, the ADFS allows the setting up of a so-called library directory and drive (using the *LIB command). The DFS has a similar feature but it is of limited use. If the ADFS fails to find a particular filename preceded by an asterisk, it will automatically search the designated library directory for it. This means that you can keep a whole set of EXEC files in a library directory, and every time that you enter the filename preceded by an asterisk, the relevant file will be found, regardless of the current directory. In fact you may even assign a library filing system with *LIBFS.

Unfortunately the identity of the library directory is not stored on the disc itself, but held in memory. This means that it must be re-assigned each

time that you change disc, or power up, and in particular, each time that *MOUNT is issued. Perhaps the easiest way to set the directory is to do it in the boot file of every disc. It is then automatically set every time that the disc is booted. Another way is to use Ctrl-A-Break. Each time that this is issued, the filing system searches the root directory of the disc for a directory beginning with the letters LIB. If it finds one, it designates this as the library, if not, it designates the root directory as the library. A third method, and there is something to be said for using a combination of all three, is to use an EXEC file to reset the library directory. This could be called \$.L, and would contain the single line:

```
*LIB $.LIBRARY or whatever
```

Thus if you found at any time that the library was unset, you would just issue the command *.L

CREATING EXEC FILES

EXEC files may be created in any one of a number of ways. Users of the Master will probably find the Editor the simplest option once they are sufficiently familiar with it. To use the Editor, enter it with EDIT from Basic, then type in your required text without line numbers, pressing Return at the end of each line. Then use f3 to save the file under the chosen filename. As an alternative, you may use View, Wordwise, or the command *BUILD (see manual). But these three methods have certain limitations, and for this reason a separate note is given on a method which relies on Basic. We now take a look at some of the many applications of this extremely useful type of file.

CONTROLLING THE PRINTER

Remembering printer control codes, or even trying to look them up in the manual can be a pain. The EXEC file solves the problem with the greatest of elegance. For example, you have only to create a file called ELITE that sets up the codes for the Elite typeface on your printer, and any time that you type *ELITE, the job will be performed, whether you are in Basic, View or wherever. By now you have probably already thought of a dozen printer codes that you could put on file in this way. The accompanying table gives one or two useful Epson codes including the very handy skip-over-perforations, and a five-space indent for use with program

SOME USEFUL EPSON CODES

Command	VDU codes	Function
*SKIP	2,1,27,1,78,1,6,3	Skip perf
*CSKIP	2,1,27,1,79,3	Cancel skip
*INDENT	2,1,27,1,108,1,5,3	Indent
*CINDENT	2,1,27,1,108,1,0,3	Cancel indent
*RESET	2,1,27,1,64,3	Reset not MX80
*ELITE	2,1,27,1,77,3	Elite face
*PICA	2,1,27,1,80,3	Pica face
*HASH	2,1,27,1,82,1,0,3	Set hash (#)
*POUND	2,1,27,1,82,1,3,3	Set pound (£)

listings so that your hole-puncher does not perforate them irretrievably. The accompanying codes in the table are sequences of VDU codes. These are best included in the file as pure control codes, rather than as direct VDU calls. That is to say it is preferable if your EXEC file contains the control character 27 rather than the call VDU 27, for example. This can be achieved using the Editor, but is probably best accomplished using the Basic method, described in the accompanying inset.

So far we have considered printer codes in isolation, but you may well wish to put several of them on a file together. A powerful example of this technique is given by a printout file which performs the following functions:

- 1 Set LISTO option e.g. LISTO 7
- 2 Set WIDTH 60, say, to give RH margin
- 3 Set printer to skip perforations
- 4 Set printer to print # in place of £
- 5 Set printer to give 5 column indent
- 6 Turn printer on and perform LIST
- 7 Perform form feed
- 8 Reset printer and turn off

To achieve all this, you just type *P, or whatever, and the whole process is carried out automatically. A program to create such an EXEC file is given as an example on the accompanying section on creating EXEC files from Basic.

AUTOMATED PATH SELECTION

The multiple nesting capability of the ADFS means that files which you are working on may well be deeply nested. The EXEC file can provide a very simple way of creating instant access to selected paths. For example, suppose that you are working on files in a directory accessed as follows:

\$.OCT-WORK.PROG-DEV.UTILITIES.filename

This is obviously a tortuous route to travel, even though it uses only four of the potential 127 nesting levels of the filing system. However, you can provide a very fast route to this directory by creating an EXEC file whose content is:

*DIR \$.OCT-WORK.PROG-DEV.UTILITIES

If you give this file the name "U", say, and save it in the library directory, then to get to the specified directory, regardless of the currently selected directory, just type:

*U

It is as simple as that.

INSTANT BACKUPS

The same principle may be used to permit the making of instant backups on a separate disc, using a dual drive as follows. Set up one or more directories on the second drive to hold the backup files. They could be named \$.PROGBACKUP and \$.TEXTBACKUP for the sake of argument. Now create an EXEC file with the following contents:

SAVE"!.\$.PROGBACKUP.

It should not be terminated with a Return (i.e. it should be on the very bottom line in the Editor, or if you are using the Basic method, put a semicolon at the end of the PRINT statement). Save this file with the name "PB" (for Program Backup) in the library directory of your work disc, which we will assume is kept in drive 0. Now, whenever you wish to backup a program, type *PB <Return> followed by the name of the file, followed by a single quote. Pressing Return will save the file in the program backup directory on the backup disc in drive 1, while leaving you in your currently selected directory on drive 0. A similar file can be used from within View, except that the two quotation marks, one in the EXEC file and the other to terminate the filename, are not necessary, and the word PROGBACKUP would be replaced with TEXTBACKUP.

A similar process can be achieved with Wordwise Plus, except that here, the EXEC file can even supply the previously used filename on which to save the backup. The following EXEC file saved under the name TB in the library directory of your work disc will achieve this:

:PRINT"Current Filename: "+F\$

:SAVE TEXT"!.\$.TEXTBACKUP."+F\$

Again there should be no Return after this

CREATING EXEC FILES USING BASIC

Essentially this method of creating EXEC files uses a Basic program to spool out the required file. Anything that is to appear in the file, except VDU calls, should appear within a PRINT statement in the Basic program. You may choose to save away the original Basic program for future editing, since it will not be easy to edit the EXEC file itself.

As an example, the following Basic program will produce an EXEC file called "p60" which performs a formatted program printout for the Epson FX80 as described in the section on "Controlling the Printer".

```
100 REM CREATES PRINTER FILE v 5
110 *FX3,4
120 *SPOOL p60
130 PRINT"*| Program Printer"
140 PRINT"LISTO 7"
150 PRINT"WIDTH 60"
```

```
160 VDU 2,1,27,1,78,1,6:REM SKIP PERF
170 VDU 1,27,1,82,1,0 :REM SET HASH
180 VDU 1,27,1,108,1,5 :REM INDENT
190 PRINT"LIST"
200 VDU 1,12 :REM FORM FEED
210 VDU 1,27,1,64,3 :REM RESET
220 PRINT"LISTO 1"
230 PRINT"WIDTH 0"
240 *SPOOL
250 *FX3,0
```

The *SPOOL commands at the start and end of the file create the EXEC file, which will print the title to the screen, and then set a width of 60, set LISTO 7, then pass various codes to the printer before listing the program. At the end of the listing, the printer is reset (using Esc 64, not implemented on the MX80). The two *FX3 calls are to prevent the printer being sent form feeds when you run the program to create the EXEC file. After you have run the program once, typing *P60 will call in the EXEC file and initiate printout.

last line. Executing *TB from within Wordwise Plus causes the last-used filename to appear on-screen. If you wish to save the backup with the same filename, just press Return. If you wish to alter the filename, you can delete the "+F\$" appearing at the end of the command line on-screen and replace it with the name required. F\$ is a variable which holds the last-used filename in Wordwise Plus. Once the save has been performed, you are returned to Wordwise with the current filename, and the currently selected directory unchanged. And if you want to catalogue the backup directories, further EXEC files of the form:

*CAT :1.\$\$.PROGBACKUP
can be used.

INSTANT INFO

If you require instant on-screen information on a particular subject you can store this in an EXEC file as follows. Suppose that you need to remember the addresses of the machine's various system calls, then you can create an EXEC file as follows:

```
*| SYSTEM CALL ADDRESSES
*| OSCLI &FFF7
*| OSBYTE &FFF4
*| OSWORD &FFF1 etc
```

Save the file in the library directory of your work disc under the name CALLS, and every time that you get stuck for a call, just type *CALLS, and the help message will be listed to the screen. This technique can usefully be employed to list out the names and functions of your major EXEC files, perhaps using "H" as the filename. But you should note that the *BUILD method of creating EXEC files cannot be used because it does not accept the "|" characters.

HEADERS AND FOOTERS

If you frequently use headers in View or Wordwise for setting up formats or rulers or whatever, you can easily EXEC these in. In Wordwise Plus, you could create a file such as the following

```
:CURSOR TOP
:LOAD TIC "$.LIBRARY.header"
```

The effect of this when called from within Wordwise Plus is to send the cursor to the top of the text area, and then load to the cursor (without destroying any resident text) the Wordwise file called "header" which is stored in the \$.LIBRARY directory. A similar process can be used for any command sequence which you wish to append to the end of your text.

As you begin to think about it there are so many uses for EXEC files.



MASTER SERIES

Master
Hints

Another crop of hints for the Master and Compact rounded up by David Graham.

EASY DFS CAT FROM ADFS

To catalogue a DFS disc in drive one, when you are currently using the ADFS on drive zero, just type:

```
*-DISC-.1
```

CO-RESIDENT EDITOR AND PROGRAM

It is sometimes useful to be able to call the Master's Editor without losing the currently resident Basic program. This may be achieved as follows. It assumes that you have a program resident at PAGE=&E00, which is less than 16k in length:

```
*FX180,&50 <Return>
```

```
Press Break
```

```
EDIT <Return>
```

```
Use the Editor
```

```
<Shift><f4> BASIC <Return>
```

```
PAGE=&E00 <Return>
```

And you have your original program back. The output from the Editor is left in memory at &5000. If you type PAGE=&5000 at any time, this will retrieve your edited program. If you wish to increase the size of the Editor buffer, at the expense of the length of your original program, simply reduce the FX call parameter (currently at &50, giving a OSHWM setting of &5000).

EXECUTIVE EDITOR

The following EXEC file, which may itself be created in the Editor, and saved in the library directory under the name "E", will set three function keys and then call the Editor from immediate mode Basic. The key settings are as follows: <f0> calls the Editor from Basic, <f4> is used to return to Basic from within the Editor, and to set mode 135 (use Shift-<f4> then Shift-Ctrl-<f4>), while <f1> is used to reverse the foreground and background colours in modes 0 and 3 within the Editor. This can make text more easily readable, when the resolution of the

monitor is not all that it might be. This key works equally well within or outside the Editor. To use it from inside the Editor, press <f1> then Shift Ctrl <f1>.

```
*FX 228,1
```

```
*KEY0 EDIT|M
```

```
*KEY1 |S|e|G|e|@|e|S|A|e|@|e|@
```

```
*KEY4 BASIC|M MODE 135|M
```

```
EDIT
```

A further bonus of using mode 3 in the Editor with reversed text is that the whole screen contains ruled lines, making certain types of editing work a little easier. One last point on the subject of the Editor: it is excellent for producing EXEC files like the one above, but remember to use the internal save and load options, rather than attempting to return to Basic, which destroys any text without line numbers.

EDITOR PROGRAM LOSS

If you have a Basic program in memory, and decide that you just want to edit a few lines of it, say from line 10 to 50, avoid the temptation to issue the command:

```
EDIT 10,50
```

It is perfectly legal, and positively tempts you to use it. But when you use Shift <f4> to return to Basic, the newly edited lines will be ALL that remains. The rest of the program will have been overwritten.

ADFS *ACCESS

The various ADFS access parameters do not function quite as the Master manuals appear to imply for LOAD and SAVE operations. The accompanying table is an attempt to clarify the situation.

STATUS	LOAD	SAVE
R or RW	Yes	Yes
LR or LWR	Yes	No
W or None	No	Yes
L	No	No

From the information supplied in the manuals, one would expect the attribute R to allow read-only (as it does in the case of OPENUP/OPENIN/OPENOUT), whereas to achieve this you need to use LR or LWR. The "None" in the third row is an unexpected option. It is achieved quite simply by calling *ACCESS with no parameters after the filename, and has the same effect on LOAD and SAVE operations as the W attribute.



MASTER SERIES

Master Font
Manager

**Fed up with pica on
your monitor? Brian
Knott shows how to
swap fonts with ease
on the Master or
Compact.**

One of the differences between the Model B and the Master series is that the storage area for the function key definitions and the user-defined character sets has been moved from &B00-&CFF to the Master's Private RAM. One consequence of this is that users can no longer directly *SAVE or *LOAD function key definitions or user-defined characters. BEEBUG Vol.5 No.2 included an article on saving and restoring function key definitions - this article shows how to do the same for user-defined character sets.

The currently defined character set is held in the Private RAM locations &8900 to &8FFF - these locations cannot be accessed directly because the current language ROM is paged into memory at &8000. However, the Master's designers have provided a flag which, when set, gives access to the Private RAM at locations &8000 to &8FFF. The flag is at location &FE30 and its use is documented in the Reference Manual Part One, Section F2. Whilst the flag is set, it is possible to read from, or write to, Private RAM, but whilst we do this we cannot have a language ROM active, so we must work in Assembler.

SAVING A CHARACTER SET

The technique used here for saving a character set is to switch in the Private RAM, copy all the character definitions into main memory, switch out the Private RAM, and then *SAVE the area of main memory. Program 1 does this - type it in and save it as "SAVEFONT" (or, say, "SFONT" if you are using the DFS). To use it you must first define your characters, either manually or with a character definer, then CHAIN "SAVEFONT". The program will prompt you for a filename to use for the saved character set. It assembles the machine code at location

&DD00 (the 'transient program area' advised by Acorn for holding short machine code) and temporarily stores the full character set at locations &7500 to &7C00 in main memory. This temporary store is the top &6FF bytes of mode 7 memory.

Program 1 - SFONT

```

10 REM Program SAVEFONT
20 REM Version B 0.43
30 REM Author Brian Knott
40 REM BEEBUG November 1986
50 REM Program subject to copyright
60 REM :
100 MODE 7
110 PROCAssemble
120 CALL savechars
130 REPEAT
140 PRINT"CHARACTER FONT SAVE ROUTINE"
''
150 INPUT"Name for file :- ",name$
160 UNTIL name$<>""
170 OSLCI "SAVE "+name$+" 7500+6FF"
180 END
190 :
1000 DEFPROCAssemble
1010 LOCAL C%
1020 FORC%=0TO 2 STEP 2
1030 P%=&DD00
1040 [OPT C%
1050 .savechars
1060 LDA #&80 \ set flag to
1070 STA &F4 \ access Private
1080 STA &FE30 \ RAM
1090 LDA #&FF \ &70,&71 hold num
1100 STA &70 \ of bytes to move
1110 LDA #6
1120 STA &71
1130 LDA #&89 \ &72,&73 hold
1140 STA &73 \ origin location
1150 STZ &72
1160 LDA #&75 \ &74,&75 hold
1170 STA &75 \ destination locn
1180 STZ &74
1190 .mover LDX &71:LDY #0
1200 .loop LDA (&72),Y:STA (&74),Y
1210 DEY:BNE loop:INC &73
1220 INC &75:DEX:BMI end
1230 BNE loop:LDY &70:BNE loop
1240 .end LDA #12 \ back to BASIC
1250 STA &F4:STA &FE30:RTS
1260 ]
1270 NEXT:ENDPROC

```

LOADING A CHARACTER SET

The technique for loading a character set is similar but reversed. Type in Program 2 and save it as "LOADFONT" (or, say, "LFONT" if you are using the DFS). To use it, just CHAIN "LOADFONT". It will

Program 2 - LFONT

```

10 REM Program LOADFONT
20 REM Version B 0.43
30 REM Author Brian Knott
40 REM BEEBUG November 1986
50 REM Program subject to copyright
60 REM :
100 MODE 7
110 PROCassemble
120 REPEAT
130 PRINT"CHARACTER FONT LOAD ROUTINE"
"
140 INPUT"Name of file to load :- ",na
me$
150 UNTILname$<>""
160 OSCLI "LOAD "+name$+" 7500"
170 CALL loadchars
180 END
190 :
1000 DEFPROCassemble
1010 LOCAL C%
1020 FORC%=0TO 2 STEP 2
1030 P%=&DD00
1040 [OPT C%
1050 .loadchars
1060 LDA #&80 \ set flag to
1070 STA &F4 \ access Private
1080 STA &FE30 \ RAM
1090 LDA #&FF \ &70,&71 hold num
1100 STA &70 \ of bytes to move
1110 LDA #6
1120 STA &71
1130 LDA #&89 \ &72,&73 hold
1140 STA &73 \ destination locn
1150 STZ &72
1160 LDA #&75 \ &74,&75 hold
1170 STA &75 \ origin location
1180 STZ &74
1190 .mover:LDX &71:LDY #0
1200 .loop:LDA (&74),Y:STA (&72),Y
1210 DEY:BNE loop:INC &73
1220 INC &75:DEX:BMI end
1230 BNE loop:LDY &70:BNE loop
1240 .end:LDA #12 \ back to BASIC
1250 STA &F4:STA &FE30:RTS
1260 ]
1270 NEXT:ENDPROC

```

prompt you for the filename of the character set, will *LOAD the definitions into main memory, switch in the Private RAM, copy the characters across, and then

switch out the Private RAM. When you change to a graphics mode, the new character set will be evident.

CHARACTER SETS ON THE WELCOME DISC

The TIMPAINT program on the Welcome Disc includes three full character sets - Italic, Thin, and Wide. These sets can be individually loaded into the Master and then saved using SAVEFONT. They may then be re-loaded as required using LOADFONT. Assuming that you have the Welcome Disc in the drive, to load the Italic set you should type "\$.LIBRARY.ITALIC", and to load the Thin set you should type "\$.LIBRARY.THIN". The Wide set needs to be set up from inside the Timpaint program, so first CHAIN "\$.TIMPAINT", then select font Fc using the pointer and the Ctrl key, then press Break. Having loaded any of these sets you can now change the disc and CHAIN "SAVEFONT" to save it.

Also on the Welcome Disc is the CHARDES program which will allow you to define and save your own character sets. Any sets saved in this way can be loaded in by *RUNning the saved file.

CHANGING CHARACTER SETS IN A BASIC PROGRAM

It is possible to load different character sets from within a Basic program, using LOADFONT as a basis, as long as you work in a SHADOW mode (modes 128 to 135), and as long as the memory required by your program does not go beyond &74FF.

You should include the assembling procedure (lines 1000 to 1270) in your program and call this Procedure at the start to set up the code. Then, assuming you have used SAVEFONT to save a set of characters called ITALIC, you only need the following two lines to change to Italics:

```

*LOAD ITALIC 7500
CALL &DD00

```

If you are in a graphics mode, this will immediately change to the Italics character set, otherwise the new characters will be seen when you select a graphics mode.

27 There are ten different types of monster, all drawn in large-scale multi-coloured graphics, and an even greater number of attacking patterns.

Galaforce is professionally presented,

with a demo mode, excellent sound effects and graphics. Even though Galaforce wins few marks for originality it is very enjoyable to play. Superior have managed to keep up their growing reputation as one of the Beeb's leading software houses.

DIGITAL STORAGE OSCILLOSCOPE

(Part 2)

This month, David Peckett puts the finishing touches to his oscilloscope program, adding a variety of routines, including those for saving and loading data.

This month's listing completes the storage 'scope program begun last month by implementing four additional procedures. These are concerned with the Load, Save, Star command, and Time Scale selection options, present only in dummy form in part one of the program.

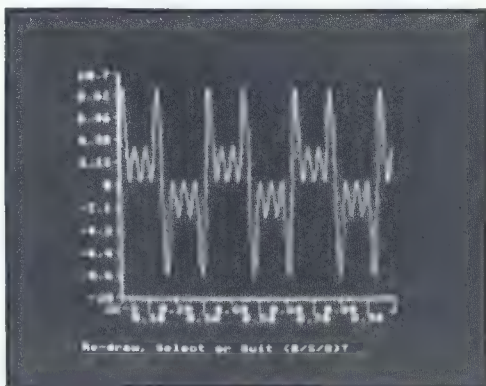
To enter these additions, which overwrite the four dummy procedures at the end of last month's listing, you may either load in last month's program, and then type in the current listing, or type in this month's listing separately, and then merge it with last month's. The merge operation could be variously achieved, either using Toolkit's *MERGE, the merge utility given elsewhere in this issue, or by using *SPOOL and *EXEC as described in Printer Graphics (part 3), again in this issue.

THE NEW OPTIONS

When you run the full program, the menu will look the same as before, except that the Save, Load ('Old data') and Star command options will now function correctly, though you should note that the Save option does not appear on the menu until there is data to be saved. You may also notice that the Star command option contains a trap for commands such as *COPY which would overwrite the program.

The Load and Save options are very straightforward. Both request a filename before getting on with the job, and in both cases, a dynamic display of the number of bytes left to save/load appears on-screen.

The only other addition is the Select option which appears once data has been plotted. Pressing 'S' from the plot screen activates two vertical bars which may be



moved using R or L together with the left and right cursor keys. The part of the display between these bars is selected for re-plotting when D (for Draw) is pressed. In this way the user may examine in detail small time segments of the plot. The process may be repeated to further amplify the x-axis. Pressing 'Q' returns you to the main menu as before.

PLOTTING MATHEMATICAL FUNCTIONS

As suggested earlier, it is possible to use this program to plot mathematical formulae as well as data from the analogue port. To achieve this, very few changes are required. You first need to insert a line at 1585 to scale the horizontal axis. For example:

```
1585 Z=N%/100
```

In this case, if you are plotting 1000 points, the value of Z will range from 0 to 10, since N% holds the sample or point number during the plotting process, which ranges from 0 to 1000. You then need to replace line 1590, which normally reads in data from the analogue port. The following will cause it to plot SINE(Z):

```
1590 Data%(N%)=20000*SIN(Z)
```

The 20000 is just a scaling factor. It effectively 'amplifies' the 'signal' so that it is of the expected order of magnitude.

A little experimenting here will soon begin to reveal the many interesting possibilities that this program has to offer.

This month's magazine cassette/disc contains a full version of the storage 'scope program together with two sample data files to get you started

PROGRAM NOTES

PROCInit: The variable Maxpts sets the maximum number of samples the program can take (max=2500).

PROCMenu: Sets up top-level menu.

PROCRecord: Sets up recording parameters, validates them, confirms them, and then performs the recording.

PROCPlot: Sets up the graph plotting.

PROCPlota: Scales and draws the graph, and selects subsidiary options such as re-drawing a part of the data. The two plot procedures allow recursive calls and repeated drawing of parts of the data.

PROCLimits: Asks for the possible real-world range of the recorded data to permit sensible scales to be displayed.

PROCScale: Identifies the high and low values of the data points which will be displayed.

PROCGraph: Used to set up the display axes.

PROCDrawGraph: Actually draws the data on the screen. The variable LType is used to select a continuous line or discrete points, by means of a PLOT n,x,y approach.

PROCSelect: Puts the vertical cursors on the screen when part of the display is being selected for re-drawing.

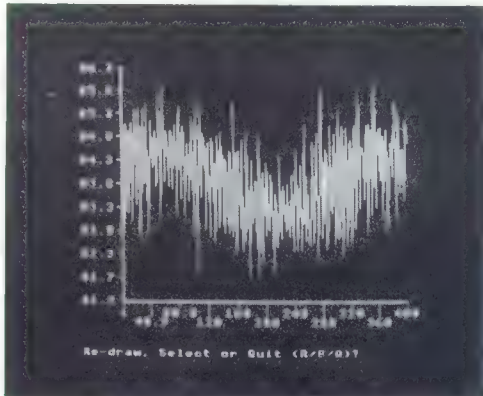
PROCSave: Saves the current sample data to disc or tape as a sequential file, displaying progress with a countdown.

PROCold: Loads data from a file, again displaying countdown. It also sets the variable hival to -1 so that the system knows that the data requires scaling.

PROCStar: Gives access to star commands, with a trap for potentially dangerous ones.

FNGetch(str\$): Used to read key presses and compare them with the range of characters in str\$. It is used with the program's menus to select options.

```
2980 REM Digital Scope Part 2 v4
2990 :
3000 DEF PROCSelect
3010 PRINT TAB(0,31) SPC35;
3020 PRINT TAB(0,31) "L/R & arrows to s
elect, D to re-Draw";
3030 XL=160:XR=1200
3040 GCOL 3,1
3050 MOVE XL,200:DRAW XL,1000
3060 MOVE XR,200:DRAW XR,1000
3070 REPEAT
3080 *FX15,0
3090 IF INKEY=87 THEN PROCCLLine
3100 IF INKEY=52 THEN PROCRLLine
3110 UNTIL INKEY=51
3120 last=INT((XR-160)/xscale+first+.5)
3130 first=INT((XL-160)/xscale+first+.5
)
3140 ENDPROC
3150 :
3160 DEF PROCCLLine
3170 MOVE XL,200:DRAW XL,1000
3180 IF INKEY=122 THEN XL=FNMin(XL+4,XR
)
3190 IF INKEY=26 THEN XL=FNMax(XL-4,160
)
3200 MOVE XL,200:DRAW XL,1000
3210 ENDPROC
3220 :
3230 DEF PROCRLLine
3240 MOVE XR,200:DRAW XR,1000
3250 IF INKEY=122 THEN XR=FNMin(XR+4,12
00)
3260 IF INKEY=26 THEN XR=FNMax(XR-4,XL)
3270 MOVE XR,200:DRAW XR,1000
3280 ENDPROC
3290 :
3300 DEF PROCSave
3310 CLS
3320 COLOUR 129:COLOUR 0
3330 PRINT TAB(15,3) "SAVE DATA"
```



```

3340 COLOUR 128:COLOUR 1
3350 INPUT ""File name to save as? "File$
3360 F%=OPENOUT(File$)
3370 PRINT TAB(10,10) "Values left to save: ";N%
3380 PRINT#F%,N%
3390 FOR I%=1 TO N%
3400 PRINT#F%,Data%(I%)
3410 PRINT TAB(31,10);(N%-I%);" "
3420 NEXT
3430 CLOSE#F%
3440 ENDPROC
3450 :
3460 DEF PROCold
3470 CLS
3480 COLOUR 129:COLOUR 0
3490 PRINT TAB(13,3) "READ OLD DATA"
3500 COLOUR 128:COLOUR 1
3510 INPUT ""Read which file? "File$
3520 F%=OPENUP(File$)
3530 INPUT#F%,N%
3540 PRINT TAB(10,10) "Values left to read: ";N%
3550 FOR I%=1 TO N%
3560 INPUT#F%,Data%(I%)
3570 PRINT TAB(31,10);(N%-I%);" "
3580 NEXT
3590 CLOSE#F%
3600 hival=-1
3610 ENDPROC
3620 :
3630 DEF PROCStar
3640 CLS
3650 COLOUR 129:COLOUR 0
3660 PRINT TAB(13,3) "STAR COMMANDS"
3670 COLOUR 128:COLOUR 1

```

```

3680 PRINT TAB(10,10);:INPUT "*"star$
3690 Lstar$=LEFT$(star$,2)
3700 OK=(Lstar$<>"CO" AND Lstar$<>"BA")
3710 OK=(OK AND Lstar$<>"L." AND Lstar$<>"LO")
3720 OK=(OK AND Lstar$<>"R." AND Lstar$<>"RU")
3730 IF NOT OK THEN VDU7:PRINT ""THAT MAY ERASE THE PROGRAM!""Do you wish to continue (Y/N)? ";:OK$=FNGetch("YN"):PRINT OK$:OK=(OK$="Y")
3740 IF OK THEN PRINT:PRINT:PROCOSCLI(star$)
3750 IF VPOS>29 THEN PRINT:PRINT
3760 PRINT TAB(0,31) "Press SPACE for main Menu ";
3770 REPEAT UNTIL INKEY=99
3780 ENDPROC
3790 :
3800 DEF PROCOSCLI(str$)
3810 LOCAL X%,Y%
3820 $osclibuf=str$
3830 X%=osclibuf MOD 256
3840 Y%=osclibuf DIV 256
3850 CALL &FFF7
3860 ENDPROC

```

ERRATUM

In last month's listing one line in the program was unintentionally duplicated. Please alter line 1580 to read:

1580 N%=N%+1

The only effect of the duplication is to slow down the reading of data. This is only noticeable at the very fastest speed setting of the oscilloscope.

```

3920 IFC%>2C%=C%*4+C%DIV8
3930 IF (C%AND2)>0FORM%=I%-Y%TOK%STEP16:
MOVEM%,L%:DRAWM%+Y%,J%:NEXT
3940 IF (C%AND4)>0FORN%=J%TOL%STEPS%:MOV
EI%,N%:DRAWK%,N%:NEXT
3950 IF (C%AND8)>0FORM%=I%-Y%TOK%STEP16:
MOVEM%,J%:DRAWM%+Y%,L%:NEXT
3960 IF (C%AND16)>0FORM%=I%TOK%STEPS%:MO
VEM%,J%:DRAWM%,L%:NEXT
3970 VDU26:ENDPROC
3980 DEFPROCline(A%,B%,P%,Q%,C%)
3990 LOCALJ%,K%:J%=240+(C%AND3)
4000 IF (C%AND4)=0K%=5ELSEK%=29
4010 GCOL0,1:VDU5:MOVEA%,B%:PLOTK%,P%,Q
%
4020 MOVEA%+8,B%+8:GCOL0,0:VDU243,8:GCO
L0,1:VDUJ%
4030 MOVEP%+8,Q%+8:GCOL0,0:VDU243,8:GCO
L0,1:VDUJ%
4040 VDU4:ENDPROC
5000 DATA Demonstration Data
5001 DATA BBC Micro Business Graphics

```

```

5002 DATA Trend for 1986
5003 DATA Sales Performance (E'000),0,1
5004 DATA Jan,1,Feb,1,Mar,1
5005 DATA Apr,1,May,1,Jun,1
5006 DATA Jly,0,Aug,0,Sep,0
5007 DATA Oct,0,Nov,0,Dec,0,Jan,0
5008 DATA XYZ Co,1,0
5009 DATA ACME plc,1,1
5010 DATA AZX Inc,1,2
5011 DATA UniGlobe,1,5
5012 DATA x,0,0,y,0,0,z,0,0
5013 DATA 24.8,23.2,11.7,4.1995,3.2,3.4
,0,0,0,0,0,0,0
5014 DATA 25.3,22.1,16.7,12.11,6.02,0,0
,0,0,0,0,0
5015 DATA 26.8,22.83,22.16,12.5,10.8,0,
0,0,0,0,0,0
5016 DATA 39.6,29.8,29.95,17.2,12.52,4.
1,0,0,0,0,0,0,0
5017 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
5018 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
5019 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0

```



by Mitch

ADVENTURE GAMES ADVENTURE GAMES

The aim of the Graphic Adventure Creator (GAC) is to free your imagination from the coils of the programming monster and allow you to create your own full-scale adventures - complete with pretty pictures.

The GAC provides a menu of options which as well as making selections easier, help to remind the user what he should be doing next. The two main selections are the Graphic Editor and the Text Editor.

The Graphic Editor is a keyboard-driven drawing package with which you create your artistic masterpieces. A sample program is included with the GAC which shows what terrific pictures can be achieved with this package.

The usual Rectangle, Circle, Line, and Fill commands are available to speed up your drawing, and various shades of colour are easily provided. As the graphic side to this whole program is its selling point, a truth should be quickly pointed out. If I sold you Leonardo's paint brush, would you be able to draw the Mona Lisa? Sadly the answer is 'NO'.

I soon found that to draw anything worthy of inclusion in an adventure would take me a very long time indeed. Anything better than a rectangle with a red triangle on top to represent the "Palace of Fairy Nuff", was going to take me more time than that taken to write the whole adventure. This unhappy situation is mainly caused by my own lack of artistic sense, but also by the lack of a number of commands which are standard in more sophisticated art packages. To create anything half as good as the examples given will require careful forethought using a paper grid and a lot of painstaking input.

Okay, if the paintbrush is only as good as the fistful of bananas wielding it, what of the text-handling side of the package? Happily I think it's pretty good. The final game can accept complex sentence input (e.g. "Take the box and open it carefully") and a 'Quick Start' file is provided which has a host of commonly required useful words and presets with which you can get your game into a working state with a minimum of effort.

In general the method of creating your game is what you would expect. You must provide a list of nouns, verbs, room descriptions, object names etc from which the editor can produce lists with numbers allocated to each item (e.g. OBJECT 1 is a SWORD, VERB 4 is KILL). As with The Quill adventure creator (reviewed in Vol.5 No.5), the nitty gritty arrives when you have to create the logic rules for your game to follow. Somehow your game must understand what to do when a player types "KILL THE DRAGON".

To solve this problem the GAC requires you to give it a list of statements which it will use to decide what action to take for a set of player commands. These statements look like simple Basic commands and as in The Quill they include special words to do common tasks (e.g. MESS 4 means write out message number 4). To complete this part of the adventure creation is always going to be awkward for absolute beginners in the art of programming, no matter how sweetly it's wrapped, but the GAC editor does it as painlessly as is possible.

So what's the best buy? The Quill has no built-in graphics (although an add-on is promised soon) and it's slightly more complex to understand than the GAC.

The GAC has sacrificed some of the Beeb's precious memory for the dubious advantage of adding pictures to your game. I believe if your aim is create fairly simple and colourful games then the GAC cannot be beaten. Coupled with the easier Logic format almost anyone can create some kind of fun game. Should you wish to create more challenging games, The Quill has the edge, as in this case the only final limit to your game's complexity is your own skill.

Surac adds to last month's date handling routines with a further set of functions and procedures providing flexible control of output.

The second and concluding part of the date handling set of routines deals with formatting the dates for output. The code for this is listed opposite. Lines 120 to 140 set up arrays containing names of days etc, and these should be included as part of your main program.

The date formatter requires a packed date (explained in part 1), plus a format string, as parameters, and returns a date as a string developed using the format string as a template. A format string is built up out of the following elements in any order:

YYYY causes the year to be inserted into the string in the form 1985. It always has four digits.

YY causes the year to be inserted in the form 85 or 05. It always has two digits and a leading zero where necessary. It is assumed that the user knows which century is being referred to!

M inserts the month as a number i.e. 12 for December or 5 for May.

MM does as above but in the case of January to September (1 to 9) it inserts a leading space as well.

MMM does as above but adds a leading zero if required, rather than a leading space.

```

120 DIM mon$(11), day$(6), mon$(11):FORa
% = 0 TO 11: READ mon$(a%), mon$(a%): NEXT: FORa%
= 0 TO 6: READ day$(a%): NEXT
130 DATA 31, JANUARY, 28, FEBRUARY, 31,
MARCH, 30, APRIL, 31, MAY, 30, J
UNE, 31, JULY, 31, AUGUST, 30, SE
PTEMBER, 31, OCTOBER, 30, NOVEMBER, 31, DEC
EMBER
140 DATA SUNDAY, MONDAY, TUESDAY
, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
1450 :
1460 DEFFNstr(u$, sel%): LOCALu$: @% = @% AND
&FFFFFF: u$ = STR$(u%): IF LEN(u$) > 1: u$ = ELSEIF
sel% = 1 THEN "0" + u$ ELSE " " + u$
1470 :
1480 DEFFNcseq(j$, q%): IF q% > LEN(j$) = ""
1490 LOCALv$, x$, y$: x$ = MID$(j$, q%, 1): REP
EAT: v$ = v$ + x$: q% = q% + 1: y$ = x$: x$ = MID$(j$, q%
, 1): UNTIL x$ < > y$: v$ = v$
1500 :
1510 DEFFNd0to6(d%): LOCALl%: l% = (FNdaydi
f(324403735, d%)) MOD 7: IF l% < 0: l% = 7 ELSE l%
1520 :
1530 DEFFNdatesp(d%, f$): LOCALw$, p%, f1$,
l%: p% = 1: REPEAT: f1$ = FNcseq(f$, p%): l% = LEN(
f1$): p% = p% + 1%: ON INSTR("DMY", LEFT$(f1$, 1
)) GOSUB 1550, 1560, 1630, 1700 ELSE GOSUB 1740
1540 UNTIL f1$ = "": w$ =
1550 RETURN
1560 ON 1% GOSUB 1580, 1590, 1600, 1610, 1620 E
LSEGOSUB 1620
1570 RETURN
1580 w$ = w$ + STR$(FNdayip(d%)): RETURN
1590 w$ = w$ + FNstr(FNdayip(d%), 1): RETURN
1600 w$ = w$ + FNstr(FNdayip(d%), 0): RETURN
1610 w$ = w$ + LEFT$(day$(FNd0to6(d%)), 3): R
ETURN
1620 w$ = w$ + day$(FNd0to6(d%)): RETURN
1630 ON 1% GOSUB 1650, 1660, 1670, 1680, 1690 E
LSEGOSUB 1690
1640 RETURN
1650 w$ = w$ + STR$(FNmonip(d%) + 1): RETURN
1660 w$ = w$ + FNstr(FNmonip(d%) + 1, 1): RETUR
N
1670 w$ = w$ + FNstr(FNmonip(d%) + 1, 0): RETUR
N
1680 w$ = w$ + LEFT$(mon$(FNmonip(d%)), 3): R
ETURN
1690 w$ = w$ + mon$(FNmonip(d%)): RETURN
1700 ON 1% GOSUB 1720, 1720 ELSE GOSUB 1730
1710 RETURN
1720 w$ = w$ + FNstr(FNyearip(d%) MOD 100, 0):
RETURN
1730 w$ = w$ + STR$(FNyearip(d%)): RETURN
1740 w$ = w$ + f1$: RETURN
1750 :

```

```

1760 DEFFNdaydif(d1%,d2%):IFd1%>d2%=FNd
aydif(d2%,d1%)*-1
1770 LOCALy1%,y2%,m%,r%:y2%=FNyearip(d2
%):y1%=FNyearip(d1%):r%=(y2%-y1%)*365:mo
n(1)=28
1780 FORM%=y1%TOy2%:r%=r%-FNlpyr(m%):NE
XT
1790 m%=FNmonip(d1%):IFFNlpyr(y1%)ANDm%
>1r%=r%-1
1800 IFm%>OFORY1%=OTOM%-1:r%=r%-mon%(y1
%):NEXT
1810 m%=FNmonip(d2%):IFFNlpyr(y2%)ANDm%
<2r%=r%-1
1820 IFm%>OFORY2%=OTOM%-1:r%=r%+mon%(y2
%):NEXT
1830 =r%+FNdayip(d2%)-FNdayip(d1%)

```

MMMM inserts the first three letters of the month i.e. December would be Dec and January Jan; remember to set up the string arrays to accomodate this feature. If capital letters are used in the array then of course you will get DEC or JAN.

MMMM inserts the entire name of the month. Caution: if you want the dates to come out in neat columns you may need to include trailing spaces in the string array elements. September is the longest name, with nine letters; a suggestion is that you make all the string array elements nine or ten characters long.

D to DDDDD does the same thing for the days as M to MMMM does for the months. Again if you want neat columns then include trailing spaces in the day names to make everything at least as long as Wednesday (which I always maintain is the longest day of the week anyway!).

Any other character, be it space, ,,:; or anything else, will be inserted as it is encountered in the format string.

Some examples:

FORMAT STRING	POSSIBLE RESULT
DDD MM YY	12 02 86
DDD.MM.YY	06. 2.09
DDDDD DD (YYYY).	WEDNESDAY 5 (2025).
YY MM DD	86 3 19

To summarise the routines and describe their use:

FNdatesp(date%,format\$) is the main function and returns a string containing the date in printable form according to

the contents of format\$.

FNd0to6(date%) returns a number between 0 and 6, representing a day of the week, to reference day\$().

FNdaydif(firstdate%,secondate%) returns an integer equal to the number of days between the two dates (hark all ye biorhythm writers). If the first date is earlier then the result is positive. If the first date is the greater then FNdaydif calls itself recursively just once with its own formal parameters reversed, then multiplying by minus one. Note that using the day strings can be slow if years far from 1986 are used, as FNdaydif has a lot of work to do.

PROCdmyout simply acts as a collection point for FNyearip, FNmonip and FNdayip and PROCfeb to prevent the need to call them all separately on various occasions.

PROCyok performs a check to ensure that year boundaries have not been violated, and sets the date to 31 December 9999 or 1 January 1752 if they have, sounding a warning as well. The variables d%, m% and y% are global to PROCdmyout and PROCyok; so exercise caution if you want to use these particular routines on their own (I don't think you will).

FNdayip(date%) strips the days part of a date from a packed date and returns it as an integer.

FNmonip(date%) and FNyearip(date%) do likewise for the month and year. The usual postfix convention applies to the function names, 'ip' referring to 'packed to integer'.

All the procedures and functions referred to must be included (the last four were listed with part 1), for date output.

The magazine cassette/disc contains a demonstration program which includes a number of useful combinations of different functions and procedures, as well as a complete trial of all the formatting possibilities. Overall, the date handlers provide a means for you to use dates easily with your own programs without the extra work imposed by writing such handlers yourself.

Watford ROM/RAM board

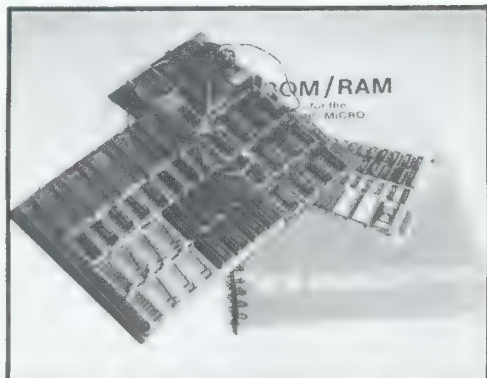
Now that ROM and RAM boards are almost 'de rigeur' for all self-respecting Beeb enthusiasts, Watford Electronics have produced an all-singing, all-dancing version for the model B. Geoff Bains, fresh from Broadway, reports.

Product : ROM/RAM Board
Supplier : Watford Electronics,
 250 Lower High Street,
 Watford.
 (0923) 377774
Price : £51.75 (32K)
 £79.35 (64K)
 £125.35 (128K)

It is difficult to believe that at this late stage in the day a ROM board can be made that has something new to offer users of the BBC micro. However, that is just what Watford Electronics has done. Their ROM/RAM board is an extension ROM board for the Beeb that offers the user far more than any previous device.

The ROM/RAM board in fact has only three ROM sockets; the rest of the board is taken up with sideways RAM. The philosophy here is to substitute RAM for ROM sockets, and in fact the board is available in a range of different models with differing amounts of sideways RAM from 32K up to 128K. All are based around the same circuit board, and the cheapest may be upgraded to the most feature-filled by plugging in a few RAM chips and, unfortunately, soldering in a few items. Unless performed by Watford itself, this will invalidate your guarantee!

In practice the ROM/RAM board is not that simple to fit. First the 6502 processor must be removed, the board fitted into its socket, and the 6502 fitted on the board. An analogous procedure is then performed on two of the power leads that plug into the Beeb's circuit board. These are disconnected and plugged into the ROM/RAM board, and two new leads are connected from the ROM/RAM board to the vacant connectors. Finally, a flying lead is plugged onto a link (S21)



on the main circuit board. Clearly, this is not without its hazards, and should be undertaken only by those who feel confident about such matters.

The ROM/RAM board covers the entire back right quadrant of the Beeb. Users of the Watford 32K shadow RAM board can still operate the ROM/RAM board with this. However, it will not work properly with an Aries B-32 board.

Once installed, the board gives you three extra ROM sockets in positions 12, 13 and 15. Positions 8 to 11 are now the four ROM sockets on the original Beeb circuit board, and these can be used as normal. Two 8K sockets are provided in position 14 for CMOS static RAM chips, although an 8K ROM can also be fitted here. Alternatively, a PCB track may be cut (!) to allow one of the two sockets in position 14 to take a 16K ROM.

There is provision for battery backup for the RAM in bank 14. This requires only the installation of a rechargeable battery. A link (or alternatively a switch) may be fitted to 'read protect' this bank. This enables you to 'remove' this RAM should it contain some software under development that has hung the machine.

The sideways RAM in sockets 0 to 7 forms the main RAM banks. In its cheapest configuration the ROM/RAM board has 32K (two sideways banks) of RAM fitted. Up to eight banks may be fitted, either at the time of purchase or later. This is all dynamic RAM, like the main RAM in the basic Beeb. It cannot be battery backed so

it loses its contents when the power is off. However, the RAM chips themselves are both smaller and cheaper than CMOS static RAM. These eight banks of sideways RAM can provide the user with a 'silicon disc' system or a printer buffer (see below).

UTILITY SOFTWARE

A disc is provided with the ROM/RAM board. This contains utilities to load and save ROM images in any specified sideways bank. The advantage of RAM over ROM for sideways software is a debatable one. RAM obviously allows 'ROMs' to be changed without taking your machine to bits, but you do have the inconvenience of loading your sideways ROM software from disc before it can be used, since only one RAM bank can be battery backed. However, what really sets this ROM/RAM board apart from other boards is the other software that accompanies the device.

First off, there is a printer buffer utility. This is similar to that provided with Watford's sideways RAM modules (see BEEBUG Vol.5 No.1), and it enables you to continue using the machine while printing. The sideways ROM buffer software is loaded into a sideways RAM bank and takes up only about 1.5K. The rest of that bank is used as an extended printer buffer. It worked well on test and, unlike earlier versions, does not automatically clear the buffer when a disc read is performed. * commands are provided to switch on and off the extended buffer and to clear it, either with Escape or another * command.

Even more impressive than the printer buffer software is the 'Silicon Filing System'. This is provided as a sideways ROM image which can be loaded into one of the banks of RAM (preferably bank 14).

The SFS uses any number of the dynamic RAM banks as a RAM disc (or 'silicon disc'). This appears to the user just like a normal disc drive, except that data and programs are actually stored in sideways RAM and not on a disc. This means that the data is lost when the power is switched off, but all read and write operations are much faster than with a normal disc drive, once the data has been loaded. A 10K Wordwise file loaded using the SFS in about 1 second - some five times faster than loading from disc.

Almost all functions of the SFS are similar to the normal Acorn DFS, but with extensions as for the Watford DFS. There are some differences. *DRIVE, and the drive specification in other commands, has no effect in the SFS - only one RAM 'drive' is provided. If *ENABLE is omitted then commands such as *DESTROY issue a 'Go (Y/N)' prompt. The RAM disc has a default title of 'Silicon disc'. All SFS workspace is in sideways RAM and so PAGE is not moved by enabling the SFS.

Five extra commands are provided too. *SILICON initiates the SFS as the current filing system (though the manual is slow to inform of this). It also checks to see which dynamic RAM banks are free for use with the SFS. The SFS can also be entered with S-Break. *INIT may be used to specify certain banks only for the RAM disc. *CFSDISK and *CTSDISK are provided for copying files between the RAM disc and the current filing system. Unfortunately, the full use of wildcards is not permitted in these commands.

Finally, *RLOAD is provided for easy loading of sideways ROM images to sideways RAM. Suitable error messages such as 'Socket in use' and 'Not RAM' are issued when necessary with this command.

To my mind it would have been better to implement the RAM disc as another disc 'drive' (number 4) rather than as a new filing system. However, the SFS is a very powerful piece of software. Choosing to implement the RAM disc just like the normal DFS is a masterful touch. It means that using the SFS requires little learning for most users. It is a pity that Watford has taken this to the extreme by omitting any explanation of the majority of the SFS commands. Instead, users are referred to the Watford DFS manual - a cheap trick.

However, the ROM/RAM board provides a new approach to sideways ROM boards that will appeal to users who prefer sideways RAM to large numbers of ROM sockets. The board itself is well thought out and constructed and the software both useful and well written. This device is expensive but if it provides the facilities which you are looking for, then in my view it represents good value.



POSTBAG

The Red Sea Scrolls

My thanks to BEEBUG and to Graham Blackwell for the excellent 'filler' program and the British Isles map (see BEEBUG Vol.5 Nos.3 & 4). However, could you suggest why, after running the program to get a green Britain on a blue sea and saving the screen to disc, on reloading directly to the screen using:

```
MO.2:*LO.<filename>
I get a sea of RED?
```

A.Jepson

This effect results from the way in which the Beeb handles 'logical' and 'actual' colours. In any mode, the colours are referenced by number and are linked to a default set of colours as described in the User Guide. However, the colours in use at any time can be changed using VDU19, and this is what happens to produce a green Britain with a blue sea. When the screen is saved, the colour information is in 'number' form only. When the screen is re-loaded as described, it is the default colours that are used, in this case substituting red for blue.

If you want to re-load the screen with the 'correct' colours, then these will need to be set as follows:

```
MO.2:VDU19,1,4,0,0,0
*LO.<filename>
```

It is worth noting that the colours in a screen display can be changed after loading (without corrupting the picture) by typing Ctrl-S (instead of VDU19) followed

by the 5 digits (without commas) that usually follow a VDU19 command.

Rheumy-eyed but Trendy

As an old (in every sense!) BEEBUG member, I have every copy of the magazine.

I was irritated but silent when you started putting the supplement inside the main magazine. I am peeved but resigned now that you have made it part of the magazine.

However, I will NOT remain silent about your extraordinary decision to put Hints and Tips onto a page covered in the word 'HINTS'. Maybe you all have excellent eyesight; maybe as you know what is on the page you haven't tried to read it. I haven't, I don't, I have, and I can't!

I'm sure it seemed a good idea at the time, but you don't have to be particularly rheumy-eyed to find this style of printing difficult to read. I look forward to future clear, if less trendy, magazines.

Yours, astigmatically,
Mrs Gill Turner

Mrs Turner, and other readers who feel similarly, will be pleased to see that we have indeed become less 'trendy'. We thought that this presentation would enable the hints to be more readily distinguished from the Postbag page opposite. It didn't work sufficiently and Mrs Turner's letter

confirmed our own misgivings; hence the change.

Passing On

With reference to the article on 'Passing Arrays to Procedures' (see BEEBUG Vol.5 No.2), this appears to pass arguments (i.e. array elements) one at a time only. Is there a way (1) to pass the whole array over, and (2) to pass arrays of real numbers, not just integers.

Bob Ames

In fact, the technique described in the article, did involve passing a pointer to an array as a parameter to a procedure, and using indirection operators for any processing. Particularly when dealing with large arrays, it is undesirable to literally move or transfer an entire array within memory because of the time taken, and the need for double the amount of memory space. Instead, as in the example programs, merely a pointer is passed to the procedure, and this is used to access the array where it is already located.

Indirection operators only provide directly for byte and integer (four byte) operations. To implement real arrays (five bytes per number) would therefore require the user to write his own routines to do this. It would then be quite feasible to pass a pointer to a real array as a parameter and to process this as required.

HINTS HINTS HINTS HINTS HINTS

Graphics Turn On

The Graphics ROM from Computer Concepts is usually enabled by entering *FX162 and then pressing Break. A simpler method is to press Break between the two beeps (one long and one short) that are produced when switching on the Beeb. This can be tricky to time correctly but automatically activates the ROM.

Gareth Letshon

ADFS File Names

Full stops should be avoided in filenames used with the ADFS. Although a filename such as 'VIEW2.1' is quite acceptable to the standard DFS, in the ADFS this would be treated as file '1' in directory 'VIEW2'.

Richard Sterry

*EXEC Nesting

A spooled file can call another (i.e. it can contain a *EXEC command) but you can never 'return' from the second set of instructions to the first. The *EXEC should be the last command in the file to be executed.

Richard Sterry

Screen and Window Width

To calculate the width, in characters, of the screen or a text window regardless of the display mode the following short routine can be used:

```
VDU13,8
width=POS+1
```

Roger Burg

Wordwise Plus Zeros

When short of memory in Wordwise Plus the odd byte of RAM can be saved by not including any zeros in embedded commands. For example:

```
<green>TI0<white>
can be replaced with
<green>TI<white>.
```

This works with any embedded command followed by a number.

Arthur Neasden

Lives, Lives and More Lives

The number of lives in Aardvark's Frak! can be increased to 255 (or any other number) by loading the game using *LOAD Frak2 then typing ?&305B=255. Then start the game with CALL &468A.

Similarly, your chances in Micro Power's Jet Pac can be boosted with *LOAD JET-PAC, ?&302B=255, CALL &5900.

You can get 255 lives and 255 grenades in Bug Byte's Commando with *LOAD DGAME followed by ?&254D=255: ?&2553=255 and then start the game with CALL &6D00.

Peter Weston

File Data Storage

In a disc or cassette file, integers are stored in five bytes and real numbers are stored in six bytes regardless of their contents. So:

```
X=10.5:Y%=8:PRINT#C,X,Y%
stores the 8 as a five byte
```

integer and the 10.5 as a six byte real number.

```
X=10.5:PRINT #C,X,8
will store both numbers as six byte real numbers. Failure to realize this can lead to the error message 'Type mismatch'.
```

Basic can, however, read an integer value and assign it to a real variable.

Accurate Wordwise

Tearing off a sheet of paper from a printer (especially some Epson models) can result in the first line of text on the next page being printed slightly out of alignment. This can be avoided by including the following code (for Epson compatible printers except MX) as a 'green' embedded command at the top of the page or, better still, in a heading definition:

```
OC10,27,106,36
```

This performs a line feed and then a reverse feed to reset the paper position.

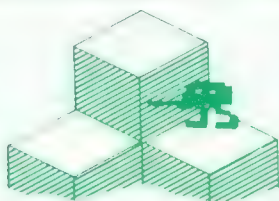
A.E. Wilmhurst

Readable Programs

Careful choice of procedure and variable names and REM statements can produce a program that reads almost like normal English. This can not only be amusing but helpful for debugging too. For example:

```
100 IF X=TRUE THEN PROC
EED ELSE PROCLAIM
200 REMOVE INVADER FROM
SCREEN
```

R.D. Bagnall



CUBERT

Can cuddly Cubert climb around the cubes cannily avoiding the crushing villain of the piece as he goes? Tim Thornham sets the scene to his colourful arcade game.

Cubert is a fast, colourful, 3 dimensional, one player game based on the popular arcade game Q*BERT. The idea of the game is to move over a 'pyramid', made up of cubes, and to change the colour of each cube. At the same time you must save Cubert from being flattened by the ball that bounces randomly down the cubes from the top of the pyramid.

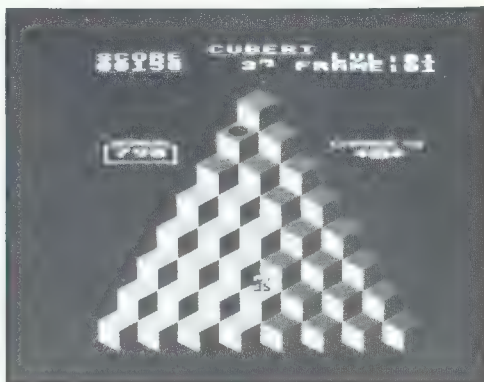
To complete a screen, all of the blocks in the pyramid need to be changed to the desired colour. Once this has been done, a new design of pyramid will appear, requiring a different strategy to change the colour of the cubes.

Once all eight pyramids have been successfully negotiated, the game restarts from the first pyramid, but each block needs to be stepped upon twice to change it to the required colour. The target colour is displayed on the screen throughout the game.

Cubert can lose a life in one of two ways, either by falling off of the side of the pyramid, or by being flattened by a bouncing ball. Cubert is given five lives to start the game with, and once these are lost, the game will end.

The quicker each screen is completed, the more bonus points are awarded. Every 5,000 points achieved brings an extra life for troubled Cubert in an attempt to reach the top of the high score table. Bonus points are displayed to the left of the pyramid, and count down as time elapses.

To control Cubert, four direction keys are needed. To move up and left (remember



you are playing in 3D) you need to press 'A'. Pressing '*' will move Cubert up and right, 'Z' moves down and left and '?' moves down and right.

If you wish to run Cubert on any system where PAGE is set higher than &E00 (e.g.DFS disc systems), then you will need to include the following movedown routine with the program. Please make sure that the program has been saved first.

```
0 IF PAGE<&E01 THEN 10
1 *TAPE
2 *KEY 0 FOR A%=0 TO (TOP-PAGE) STEP4
  :A%!<&E00=A%!PAGE:NEXT|MPAGE=&E00|M
  OLD|MRUN|M
3 *FX138,0,128
4 END
```

Please be careful when typing in the character definitions and data from line 2690 onwards, otherwise the screen display may appear garbled.

```
10 REM Program CUBERT
20 REM Version B0.1
30 REM Author Tim Thornham
40 REM BEEBUG November 1986
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO 3220
110 MODE2
120 PROCinit:PROCchars
130 REPEAT
140 FR%=1:PROChigh:PROCnew
150 REPEAT
160 IF E%=1 PROCscreen
170 VDU5:E%=1
180 REPEAT
190 PROCman
200 PROCball(0)
```

```

210 IF K>0 PROCball(1)
220 PROCman
230 IF K>1 PROCball(2)
240 PROCdelay(100)
250 PROCman
260 J%=J% EOR1:IF J%=1:PROCBonus
270 PROCscore
280 UNTIL noleft=0 OR E%=0
290 IF E%=0 PROCrestart:ELSE S%=S%+B%:
PROCscore:PROCTune
300 UNTIL L%=0
310 PROCdead
320 UNTIL FALSE
330 END
340 :
1000 DEFPROCman
1010 IF noleft=0 OR E%=0:ENDPROC
1020 IF INKEY(-66):PROCpman(-72,104):EN
DPROC
DPROC
1040 IF INKEY(-98):PROCpman(72,-104):EN
DPROC
DPROC
1040 IF INKEY(-73):PROCpman(72,104):END
PROC
1050 IF INKEY(-105):PROCpman(-72,-104):
ENDPROC
1060 PROCdelay(100)
1070 ENDPROC
1080 :
1090 DEFPROCpman(x%,y%)
1100 VDU5= SOUND2,2,50,1
1110 MOVEX%+64,Y%:PRINTCHR$127+BS(3+(PO
INT(X%-8,Y%)>9)+2*(POINT(X%+72,Y%)>9))+C
HR$(POINT(X%,Y%-64))+J$
1120 X%=X%+x%:Y%=Y%+y%
1130 W%=POINT(X%,Y%-48)
1140 IF W%<8:E%=0:ENDPROC
1150 IF W%<D:S%=S%+1:GCOL0,W%+1:MOVEX
%-36,Y%-52:MOVEX%+28,Y%-84:PLOT85,X%+28,
Y%-20:MOVEX%+36,Y%-84:MOVEX%+28,Y%-20:PL
OT85,X%+100,Y%-52:IF W%+1=D:noleft=nole
ft-1
1160 MOVEX%,Y%:PRINTM$
1170 ENDPROC
1180 :
1190 DEFPROCball(F%)
1200 IF noleft=0 OR E%=0:ENDPROC
1210 SOUND1,1,1,1
1220 MOVEB%(F%,0),B%(F%,1):GCOL0,POINT(
B%(F%,0),B%(F%,1)):PRINTCHR$250
1230 B%(F%,1)=B%(F%,1)-104
1240 P%=- (POINT(B%(F%,0)-72,B%(F%,1))>8
)-2*(POINT(B%(F%,0)+72,B%(F%,1))>8)
1250 IF P%=0 PROCrepos
1260 IF P%=1:B%(F%,0)=B%(F%,0)-72
1270 IF P%=2:B%(F%,0)=B%(F%,0)+72
1280 R%=0:IF P%=3:R%=RND(2):IF R%=1:B%(
F%,0)=B%(F%,0)-72:ELSE IF R%=2:B%(F%,0)=
B%(F%,0)+72
1290 MOVEB%(F%,0),B%(F%,1):GCOL0,0:PRIN
TCHR$250
1300 IF POINT(B%(F%,0)+32,B%(F%,1)+8)<9
:E%=0
1310 ENDPROC
1320 :
1330 DEFPROCrepos
1340 IF nos%=1:B%(F%,0)=ST%(1):ELSE B%(
F%,0)=ST%(RND(nos%))
1350 B%(F%,1)=844
1360 ENDPROC
1370 :
1380 DEFPROCrestart
1390 FORT=0T01:SOUND0,-15,T,2:NEXT
1400 PROCdelay(1000)
1410 L%=L%-1:VDU4:PRINTTAB(9,2);L%:VDU5
:IF L%=0 ENDPROC
1420 IF W%>8:MOVEX%+64,Y%:PRINTCHR$127+
BS(3+(POINT(X%-8,Y%)>9)+2*(POINT(X%+72,Y
%)>9))+CHR$(POINT(X%,Y%-64))+J$
1430 FOR F%=0 TO K:MOVEB%(F%,0),B%(F%,1
):GCOL0,POINT(B%(F%,0),B%(F%,1)):PRINTCH
R$250:PROCrepos:NEXT
1440 X%=608:Y%=256:W%=10
1450 MOVEX%,Y%:PRINTM$
1460 ENDPROC
1470 :
1480 DEFPROCBonus
1490 IF B%=0:ENDPROC
1500 VDU4:B%=B%-10:PRINTTAB(xb,yb);"00"
;TAB(xb+3-LEN(STR$(B%)),yb);B%:VDU5
1510 ENDPROC
1520 :
1530 DEFPROCdead
1540 VDU4,28,3,19,16,15,12,26
1550 PROCdbl("GAME OVER",1,3,508)
1560 PROCdbl("Press <SPACE>",1,3,444)
1570 REPEAT:UNTIL GET=32
1580 ENDPROC
1590 :
1600 DEFPROCscore
1610 VDU4:PRINTTAB(6-LEN(STR$(S%)),2);S
%
1620 IF S%>I%:I%=I%+5000:L%=L%+1:PRINT
TAB(9,2);L%
1630 VDU5
1640 ENDPROC
1650 :
1660 DEFPROCChigh
1670 PP=0:VDU4:CLS
1680 FORS=0T07:PROCdraw(S*144,800):PROC
draw(S*144,592):NEXT
1690 FORS=0 T01:PROCdraw(S*144+72,696):
PROCdraw(S*144+792,696):NEXT
1700 PROCdbl("CUBERT",1,3,784)
1710 IF S%>HI%(8) PROCchange:PP=1
1720 PROCdbl("HIGH SCORES",1,3,500)
1730 FORT=1T08
1740 PRINTTAB(2,T+18);T;" ";STRING$(10,
".");TAB(4,T+18);H$(T);TAB(15,T+18);HI%(
T)
1750 NEXT

```

```

1760 IF PP=1 PROCin
1770 VDU4:PRINTTAB(0,28);SPC(45);
1780 PROCdb1("<SPACE> to play.",1,3,99)
1790 REPEAT:UNTIL GET=32
1800 ENDPROC
1810 :
1820 DEFPROCchange
1830 FORT=7TO0 STEP-1
1840 IF S%<HI%(T):P=T+1:T=0
1850 NEXT
1860 IF P=8:GOTO 1880
1870 FORT=7 TO P STEP-1:HI%(T+1)=HI%(T)
:H$(T+1)=H$(T):NEXT
1880 HI%(P)=S%:H$(P)=STRING$(10,".")
1890 ENDPROC
1900 :
1910 DEFPROCin
1920 PROCdb1("Please enter name",1,3,99
)
1930 PRINTTAB(4,P+18);
1940 *FX15,0
1950 X%=0:Y%=0:D:A%=0:1&D00=&200A0A00:??&
D04=127:CALL&FFF1
1960 H$(P)=LEFT$(S&A00,10)
1970 ENDPROC
1980 :
1990 DEFPROCtune
2000 FORT=1TO9
2010 SOUND1,-15,120+T*4,2
2020 SOUND2,-8,80+T*4,2
2030 NEXT
2040 ENDPROC
2050 :
2060 DEFPROCdelay(TT%)
2070 FORT=1TOTT%:NEXT
2080 ENDPROC
2090 :
2100 DEFPROCcubes
2110 nos%=0
2120 GCOLOR,6:FORT=1TO100:PLOT69,RND(128
0),RND(900):NEXT
2130 READT,noleft,xb,yb,xc,yc
2140 FORU=1 TO T
2150 READA$
2160 xc%=72*EVAL("&"MID$(A$,1,1))
2170 yc%=104*EVAL("&"MID$(A$,2,1))
2180 n%=EVAL(MID$(A$,3))
2190 IF yc%=728:FORW=1 TO n%:ST$(nos%+W
)=xc%+104+144*(W-1):NEXT:nos%=nos%+n%
2200 FORV=0 TO n%-1
2210 PROCdraw(xc%+144*V,yc%)
2220 NEXT:NEXT
2230 ENDPROC
2240 :
2250 DEFPROCdraw(x%,y%)
2260 FORT=1TO5
2270 VDU19,T+10,EVAL(MIDS(CS((FR%-1)MOD
8),T,1)),0,0,0
2280 NEXT
2290 GCOLOR,12

```



```

2300 MOVE68+x%,32+y%:MOVE132+x%,y%:PLOT
85,68+x%,y%+96:PLOT85,132+x%,y%+64
2310 GCOLOR,11
2320 MOVE140+x%,y%:MOVE140+x%,y%+64:PLO
T85,204+x%,y%+32:PLOT85,x%+204,y%+96
2330 GCOLOR,13
2340 MOVE68+x%,y%+100:MOVE132+x%,y%+68:
PLOT85,132+x%,y%+132:MOVE140+x%,y%+68:MO
VEXx%+140,y%+132:PLOT85,204+x%,y%+100
2350 ENDPROC
2360 :
2370 DEFPROCnew
2380 L%=5:S%=0:LVL%=0:FR%=0:D%=14
2390 E%=1:K=0:W%=10:I%=5000
2400 ENDPROC
2410 :
2420 DEFPROCscreen
2430 CLS:FR%=FR%+1
2440 LVL%=(FR%+7)DIV8
2450 IF (FR%-1)MOD8=0 RESTORE
2460 K=(LVL%-1)DIV2:IF LVL%>5 K=2
2470 D%=(LVL%-1)MOD2+14:IF LVL%>5 D%=15
2480 PROCdb1("CUBERT",1,3,1020)
2490 VDU4:PRINTTAB(1,1);"SCORE";TAB(1,2
);"00000";TAB(14,1);"LVL:00";TAB(20-LEN(
STR$(LVL%)),1);LVL%;TAB(12,2);"FRAME:00"
;TAB(20-LEN(STR$(FR%)),2);FR%;TAB(9,2);L
%:PROCscore
2500 MOVE640,970:PRINTM$
2510 PROCcubes
2520 X%=608:Y%=256
2530 FOR F%=0TO2:PROCrep0s:NEXT
2540 GCOLOR,D%:MOVE572,204:MOVE636,172:P
LOT85,636,236:MOVE644,172:MOVE636,236:PL
OT85,708,204
2550 MOVEX%,Y%:PRINTM$
2560 MOVEB$(0,0),B$(0,1):GCOLOR,0:PRINTC
HRS250
2570 GCOLOR,3:MOVExb,yb:VDU230,231,232:M
OVExb,yb-24:DRAWxb-32,yb-24:DRAWxb-32,yb
-80:DRAWxb+216,yb-80:DRAWxb+216,yb-24:DR
AWxb+192,yb-24

```



```

2580 MOVExc,yc:VDU233,234,235,236,237:M
OVExc+74,yc-36:VDU238,9,239:MOVExc+138,y
c-36:GCOL0,D%:VDU244
2590 xb=(xb+24)DIV64:yb=(1056-yb)DIV32
2600 B%=1000:J%=0
2610 ENDPROC
2620 :
2630 DEFPROCdbl(A$,C1,C2,H)
2640 VDU5:GCOL0,C1
2650 MOVE640-32*LENA$,H:PRINTA$
2660 GCOL0,C2:MOVE656-32*LENA$,H-8:PRIN
TAS:VDU4
2670 ENDPROC
2680 :
2690 DEFPROCchars
2700 VDU23,224,0,0,0,0,56,120,240,224
2710 VDU23,225,212,22,19,17,119,119,0,0
2720 VDU23,226,0,60,126,86,70,6,14,0
2730 VDU23,227,60,66,129,169,129,1,1,30
2740 VDU23,230,0,0,247,148,228,148,247,
0
2750 VDU23,231,0,0,165,181,189,173,165,
0
2760 VDU23,232,0,0,47,40,47,33,239,0
2770 VDU23,233,0,0,244,132,135,132,244,
0
2780 VDU23,234,0,0,189,165,189,165,165,
0
2790 VDU23,235,0,0,47,168,235,105,47,0
2800 VDU23,236,0,0,120,64,112,64,120,0
2810 VDU23,237,0,0,239,73,73,73,79,0
2820 VDU23,238,8,12,126,127,126,12,8,0
2830 VDU23,239,16,48,126,254,126,48,16,
0
2840 VDU23,240,240,240,240,240,240,224,
192,128
2850 VDU23,241,15,15,15,15,15,7,3,1
2860 VDU23,242,0,0,0,0,0,24,60,126
2870 VDU23,243,255,255,255,255,255,255,
0,0
2880 VDU23,244,255,255,255,255,255,255,
255,255
2890 VDU23,250,60,118,251,253,255,255,1
26,60
2900 C$=CHR$18+CHR$0
2910 H$=C$+CHR$11+CHR$240
2920 I$=C$+CHR$12+CHR$241
2930 J$=CHR$242+CHR$8+CHR$10+CHR$243

```

```

2940 B$(0)=H$+CHR$8+I$+CHR$8+C$
2950 B$(1)=I$+CHR$8+C$
2960 B$(2)=H$+CHR$8+C$
2970 B$(3)=C$
2980 M$=C$+CHR$3+CHR$226+CHR$8+C$+CHR$0
+CHR$227+CHR$8+C$+CHR$1+CHR$224+CHR$8+CH
R$10+CHR$225
2990 ENDPROC
3000 :
3010 DEFPROCinit
3020 DIM B$(3),B%(2,1),ST$(7),C$(7),HI%(
8),H$(8)
3030 ENVELOPE1,0,0,0,0,0,0,126,-8,-7,
-7,126,30
3040 ENVELOPE2,2,0,0,0,0,0,127,-10,-5
,-2,126,0
3050 RESTORE3190
3060 HI$(0)=1000000
3070 FORT=0TO7:HI%(T+1)=1500+500*(7-T):
H$(T+1)="Beebug":READC$(T):NEXT
3080 VDU23:8202;0;0;0;
3090 ENDPROC
3100 :
3110 DATA8,35,132,704,896,704,008,117,2
26,335,444,553,662,771
3120 DATA12,25,132,704,896,704,206,117,
222,A22,331,B31,441,A41,551,951,662,771
3130 DATA12,23,132,800,896,800,602,513,
222,A22,132,B32,242,A42,551,951,662,573
3140 DATA8,23,132,480,896,480,404,513,6
22,731,642,553,464,375
3150 DATA13,39,64,320,960,312,206,315,4
24,335,241,642,C41,153,953,063,A63,172,B
72
3160 DATA12,35,552,576,480,420,404,315,
222,A22,132,B32,042,C42,152,B52,266,375
3170 DATA16,30,64,512,960,512,404,315,2
21,622,C21,332,932,441,A41,355,261,662,C
61,171,771,D71
3180 DATA13,33,64,512,960,512,201,602,C
01,315,424,332,932,441,A41,352,952,266,3
75
3190 DATA46523,53247,65432,23154
3200 DATA73562,12543,73124,72153
3210 :
3220 MODE7:REPORT:PRINT" at line ";ERL
3230 *FX15,0
3240 END

```

POINTS ARISING POINTS ARISING POINTS ARISING POINTS

BEEBUG FILER GRAPHICS (BEEBUG Vol.5 No.2)

As this Filer option uses mode 0, some of the window and position information (in mode 3) carried over from previous Filer programs gives confusing results. To correct this change three lines as follows:

```
1360 PRINTTAB(0,27)STRING$(80,"")
```

```
20420 VDU28,0,26,79,3:w=1:ENDPROC
```

```
20440 DEF PROCwindow2:VDU28,0,31,79,28
```

The magazine cassette/disc version requires no modifications.

BEEBUG MAGAZINE is produced by BEEBUG Publications Ltd.

Editor: Mike Williams

Technical Assistant: Alan Webster

Production Assistant:

Yolanda Turuelo

Secretary: Debbie Sinfield

Managing Editor: Lee Calcraft

Additional thanks are due to

Sheridan Williams, Adrian Calcraft,

Geoff Bains, John Yale and

Tim Powys-Lybbe.

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Publications Limited.

BEEBUG Publications Ltd (c) 1986

Editorial Address

BEEBUG

Dolphin Place

Holywell Hill

St. Albans

Herts. AL1 1EX

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £40 per page, but please give us warning of anything substantial that you intend to write. A leaflet, 'Notes of Guidance for Contributors', is available on receipt of an A5 (or larger) SAE.

In the case of material longer than a page, we would prefer this to be submitted on cassette or disc in machine readable form using "Wordwise", "View", or other means, but please ensure an adequate written description of your contribution is also included. If you use cassette, please include a backup copy at 300 baud.

HINTS

There are prizes of £5 and £10 for the best hints each month, plus one of £15 for a hint or tip deemed to be exceptionally good.

Please send all editorial material to the editorial address above. If you require a reply it is essential to quote your membership number and enclose an SAE.

MEMBERSHIP RATES

Send all applications for membership, membership renewals, membership queries and orders for back issues to the membership address. All membership fees, including overseas, should be pounds sterling drawn (for cheques) on a UK bank.

MEMBERSHIP SUBSCRIPTION RATES

£ 6.90 6 months (5 issues) UK ONLY

£12.90 - 1 year (10 issues) UK, Eire, BFPO

£19.00 - Rest of Europe

£22.00 Middle East

£24.00 Americas & Africa,

£26.00 Elsewhere

BACK ISSUES

(Members only)

Vol	Single issues	Volume sets (10 issues)
1	90p	£8
2	£1	£9
3	£1.20	£10
4	£1.20	£11
5	£1.30	—

Please add the cost of post and packing as shown:

DESTINATION	First issue	Each subsequent issue
UK	30p	10p
Europe	70p	20p
Elsewhere	£1.50	50p

All overseas items are sent airmail (please send a sterling cheque). We will accept official UK orders but please note that there will be a £1 handling charge for orders under £10 that requires an invoice. Note that there is no VAT on magazines.

Back issues are for members only, so it is ESSENTIAL to quote your membership number with your order. Please note that the BEEBUG Reference Card and BEEBUG supplements are not supplied with back issues.

Membership, Back Issues &
Software Address

BEEBUG

Dolphin Place

Holywell Hill

St. Albans

Herts. AL1 1EX

Hotline for queries and software orders

St. Albans (0727) 40303

Manned Mon-Fri 9am-4.30pm

24hr Answerphone Service for Access and

Barclaycard orders, and subscriptions

St. Albans (0727) 40303

If you require members discount on software it is essential to quote your membership number and claim the discount when ordering.

Magazine Cassette/Disc

NOVEMBER 1986 CASSETTE/DISC CONTENTS

MERGE, PART-MERGE & PART SAVE — a very convenient utility for program development.

STORAGE OSCILLOSCOPE — complete program (and sample data sets) to capture and display analogue data.

THE MASTER SERIES

EXECUTIVE APPROACH — an example of the power and convenience of EXEC files.

FONT MANAGER — two programs to allow character sets to be saved and loaded on the Master.

BUSINESS GRAPHICS — the first part of this program, providing power and sophistication in the display of business data.

BEEBUG PAGE DESIGNER — all the programs to design and print pages of mixed graphics and text.

BEEBUG WORKSHOP — all the routines plus a complete demonstration of data handlers.

FIRST COURSE — examples of character formation and processing.

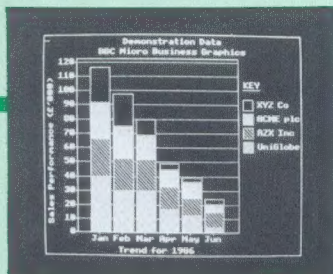
MULTI-COLUMN WORDWISE PLUS — allows you to print text in multiple columns on the page.

CUBERT — it's colourful, it's fast and it's addictive. It's Cubert, a full blooded arcade game.

BEEBUGSOFT FORUM — all the routines listed this month.

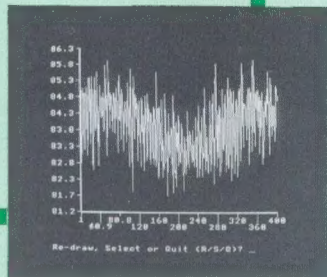
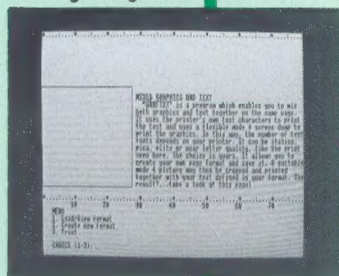
EXTRA FEATURES THIS MONTH

MAGSCAN — data for this issue of BEEBUG (Vol.5 No.6).



Business Graphics

Page Designer



Storage Oscilloscope

All this for £3.00 (cass) £4.75 (disc) +50p p&p.

Back issues (disc since Vol. 3 No. 1, cass since Vol. 1 No. 10) available at the same prices.

Subscription rates	DISC UK	CASS UK	DISC O'seas	CASS O'seas
6 months (5 issues)	£25.50	£17	£30	£20
12 months (10 issues)	£50	£33	£56	£39

Prices are inclusive of VAT and postage as applicable. Sterling only please.

Cassette subscriptions can be commuted to disc subscription on receipt of £1.70 per issue of the subscription left to run.

All subscription and individual orders to
BEEBUG, Dolphin Place, Holywell Hill, St. Albans, AL1 1EX.

DYNAMIC DISCS

FROM BEEBUG

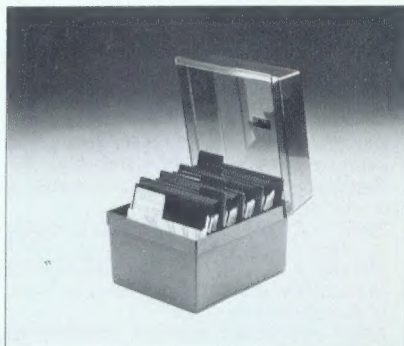
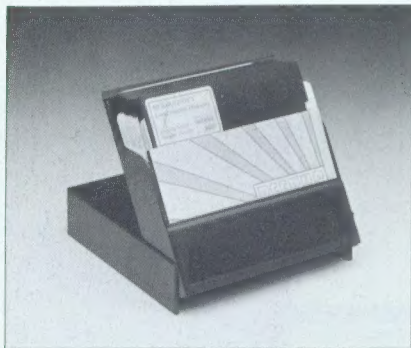
Our range of top quality 5¼" blank discs for the BBC Micro and Master 128 are now available at unbeatable prices.

These discs are from the UK's leading disc manufacturer, individually tested and guaranteed for life. They are specially produced and boxed for Beebug. Don't settle for anything less than the best.

Beebug members may claim a 5% discount on the price. If you are not yet a member please phone for an application form.

48 TPI DOUBLE DENSITY

	PRICE	MEMBERS
10 S/S	£9.90	£9.40
25 S/S	£23.00	£21.85
50 S/S	£48.00	£45.60
10 D/S	£10.42	£9.90
25 D/S	£26.47	£25.15
50 D/S	£50.00	£47.50



96 TPI QUAD DENSITY

	PRICE	MEMBERS
10 S/S	£10.42	£9.90
25 S/S	£26.47	£25.15
50 S/S	£50.00	£47.50
10 D/S	£10.90	£10.35
25 D/S	£29.00	£27.55
50 D/S	£53.00	£50.35

- Prices include FREE Storage Box
- Official Orders welcome
- ALL PRICES INCLUDE V.A.T.

- UK Postage 10's — £1.00; 25's & 50's — £3.75



ORDER FORM

Please send me _____ Qty of _____ @ £ _____ (Plus Postage £ _____)

I enclose a cheque for £ _____

or Please debit my Access/Visa No. _____

Name: _____ Membership No: _____

Address: _____

Post
Code: _____

NOTE: To claim members prices it is essential to quote membership no. If not yet a member add **£12.90** membership fee, to claim members discount.

Send to: Beebug Mail Order, Dolphin Place, Holywell Hill, St. Albans, Herts AL1 1EX
Telephone Orders welcome. Tel: 0727 40303

ISSN — 0263 — 7561